SECURITY CLASSIFICATION OF THIS PAGE

2

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

AD-A211 093

1b. RESTRICTIVE MARKINGS	
3. DISTRIBUTION/AVAILABILITY OF REPORT	Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	
5. MONITORING ORGANIZATION REPORT NUMBER(S)	AFOSR-TR- 89-0980
6a. NAME OF PERFORMING ORGANIZATION	Moore School of EE, D2 Department of SE
6b. OFFICE SYMBOL (If applicable)	
7a. NAME OF MONITORING ORGANIZATION	AFOSR
6c. ADDRESS (City, State, and ZIP Code)	University of Pennsylvania Philadelphia, Pennsylvania 19104
7b. ADDRESS (City, State, and ZIP Code)	BLDG 410 BAFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	AFOSR
8b. OFFICE SYMBOL (If applicable)	
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	AFOSR 77-3327
8c. ADDRESS (City, State, and ZIP Code)	BLDG 410 BAFB DC 20332-6448

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61102F	2304	A1	

11. TITLE (Include Security Classification)
A SIMULATION STUDY OF FOUR REAL-TIME HEURISTIC ALGORITHMS FOR MULTIPLE MISSILE MISSILE EVASION: A GAME THEORETIC APPROACH

12. PERSONAL AUTHOR(S)
M ax Mintz, Michael S. Sheketoff/ Stephan F. Huling

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
	FROM ______ TO ______		

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

AFOSR-TR- 89-0985

FINAL REPORT

Title: A SIMULATION STUDY OF FOUR REAL-TIME HEURISTIC ALGORITHMS FOR MULTIPLE MISSILE EVASION: A GAME THEORETIC APPROACH

Authors: Mr. Michael S. Sheketoff
Mr. Stephen F. Huling
Dr. Max Mintz

Date: June 1979

Submitted To: Air Force Office of Scientific Research

Prepared Under: Grant No. 77-3327

Submitted On Behalf Of: The University of Pennsylvania, Philadelphia, PA

By: Dr. Max Mintz,
Principal Investigator

Masus & Abstract

Abstract

Four real-time heuristic algorithms for determining aircraft evasion strategies against a multiple missile threat are described. Algorithms 1 and 2 are based on a "myopic" saddle-point calculation which apportions the projection of the instantaneous aircraft acceleration among the normals to the individual maneuver or guidance planes defined by each missile and its target. Algorithms 3 and 4 are also based on "myopic" saddle-point calculations. These latter two algorithms apportion the projection of the instantaneous aircraft acceleration into the individual maneuver planes so as to maximize the minimum of a function which is related to the line of sight rate of each missile threat. These latter two algorithms are motivated by the concept of anti-proportional navigation.

Each algorithm has the following properties: i) each requires relatively minimal dynamic and parametric information; ii) each provides capability against an N missile threat; iii) each generates aerodynamically feasible aircraft maneuvers which meet both structural and pilot stress limitations; iv) each is computable using foreseeable hardware; v) each exhibits markovian behavior, i.e., each is restartable from present state information.

Simulation results using each algorithm with generic F-4 and AIM-9 truth models, characterized by nonlinear differential equations, including lift, drag, gravity, 3-dimensional point mass dynamics, aircraft load factor and roll rate limits, and missile autopilot dynamics and load factor limits are presented.

All four heuristic algorithms are motivated by a formal game theoretic model for multiple missile evasion. This formal game theoretic analysis is included as part of this study.

Acknowledgment

The authors wish to acknowledge the invaluable assistance and advice of the following individuals:

Dr. H. M. Dobbins, Capt. R. N. Lutter, USAF, & Mr. M. J. Noviskey, of the Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio.

Maj. J. G. Reid, USAF, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Cmdr. F. X. Shannon, USN, & Mr. W. Weddleton, of the University of Pennsylvania.

This research was supported in part by the United States Air Force through the following grants and contracts:

The Air Force Office of Scientific Research, under Grant 77-3327;

The Charles Stark Draper Laboratory, under Contract DL-H-139457.

The technical and financial support of all of these individuals and organizations is gratefully acknowledged.

Table of Contents

List of Tables and Figures

1.0 Introduction and Summary

This report presents the results of research on the application of a blend of game theory, control theory, and decision analysis to air-to-air fire control. The focus of this study embraces the determination of aircraft evasion strategies against both single and multiple missile threats. The results include:

i) The delineation of mathematical models for the determination of optimal evasion strategies against multiple missile threats.

ii) The design and implementation of a series of computer simulation models for studying the characteristics of evasive aircraft maneuvers against both single and multiple missile threats.

iii) The determination and validation of heuristic algorithms which could be implemented to achieve real-time onboard calculation and execution of evasive maneuvers against single and multiple missile threats.

This report is presented in six parts:

Part 1. Introduction and Summary

Part 2. Overview, Alternative Problem Characterizations, and Background Information

Part 3. Heuristic Algorithms for Determining Evasion Strategies Suitable for Real-Time Onboard Computation

Part 4. Computer Simulation Models for Determining and Evaluating the Global and Local Characteristics of Evasive Maneuvers

Part 5. A Game Theoretic Model for Determining Aircraft Evasion Strategies Against a Multiple Missile Threat

Part 6. Conclusions and Recommendations for Further Research.

The material is presented in this order to emphasize the goal of applicability, as opposed to the delineation of elegant mathematical models which provide base lines or figures of merit but do not lead directly to real-time algorithms.

However, the historical evolution of this research follows the ordering Part 5, Part 4, Part 3.

The work contained in Parts 3 & 4 represent research contributions by Mr. Michael Sheketoff and are the basis for his Masters Thesis in Systems Engineering (Sheketoff, 1979). Parts 3 & 4 also provide the basis for two papers (Sheketoff & Mintz, 1979a, & 1979b). The work in Part 5 represents research contributions by Mr. Stephen F. Huling and are a part of his Ph.D. Dissertation in Systems Engineering (Huling, 1979). Part 5 also provides the basis for a paper (Huling & Mintz, 1977).

2.0 Overview, Alternative Problem Characterizations, and Background Information

2.1 Overview

The generic problem of determining how to maneuver an aircraft to evade a missile threat is by no means new. The U.S. experience in Viet Nam and the Israeli experience in the 1967 and 1973 wars have clearly demonstrated the need to develop and perfect practical methods for blunting the effectiveness of various missile threats. We cite the following open literature as being representative of recent work in missile evasion and allied studies which have been carried out by other investigators: (FAAC, 1977); (Garnell & East, 1977); (Grumman, 1975); (Grumman, 1976); (Poulter, 1975); (Shinar & Steinberg, 1976); (TASC, 1977); & (Veda, 1977). The dual to the missile evasion problem, which has aroused at least equal interest, is the determination of Missile Launch Envelopes (MLE). The missile evasion problem and its dual, the MLE determination, are clearly linked, since knowledge of when or whether to launch a missile to counter an aircraft threat depends on the potential for evasive maneuvering on the part of the target aircraft. We cite the following recent technical proposals (submitted in response to RFP F33615-77-R-1014) by the First Ann Arbor Corporation (FAAC, 1977) and the Veda Corporation (Veda, 1977) as useful sources of relevant information for characterizing the problem of missile launch envelope sensitivity analysis for an air-to-air missile engagement.

Although the dual problems of evasive maneuver determination and launch envelope determination are of interest in the air-to-air and ground-to-air contexts, we shall focus our attention on the air-to-air case in the remainder of this report. The simulation models described in Part 4 of this report could however be relatively easily modified to represent the threat of a multiple Surface-to-Air Missile (SAM) attack. In principle, the real-time

heuristic algorithms which are described in Part 3 of this report could be applied to counter a multiple SAM threat. In order to evaluate the performance of these heuristic algorithms against a multiple SAM threat, the user would need to make some modifications in the simulation programs in Part 4 to account for the appropriate dynamics and guidance of the SAMs.

2.2 Alternative Problem Characterizations and Background Information

The art of problem solving in an engineering setting requires that we choose that minimal degree of complexity and fidelity for the various models which serve to define our underlying problem, in order to:

i) capture a sufficient degree of accuracy and

ii) provide a potentially tractable solution within the confines of our computational limitations.

The determination of evasive maneuvers against single and multiple missile threats provides no exception to this general rule. In this section we shall review the qualitative and quantitative aspects of a variety of alternatives which have been proposed as approaches to obtaining a "solution" to the evasion problem associated with a single missile threat. We begin the discussion of the determination of evasion strategies by considering a single missile threat, since any evasion strategy which protects against a multiple missile threat should have reasonably good properties when used against a single missile. In the process of delineating various approaches to the single missile evasion problem, we shall refer to the dual problem, which we have referred to previously as the MLE determination. This review of current thinking about the MLE problem will provide us with a useful base line to compare such common facets as:

i) Modeling Assumptions, &

ii) Parametric Variability.

The taxonomy of missile evasion and MLE studies includes:

i) The classification of the models of the underlying dynamical systems.

ii) The classification of decision criteria which are used to measure the success of a given missile evasion scheme, or the effectiveness of a given missile engagement opportunity against a maneuvering target.

iii) Evasive maneuver modeling and generation.

The characteristics of the underlying dynamical models can be separated into two basic classes:

i) Purely Deterministic Models, &

ii) Models which include Stochastic Effects.

These two classes can, in turn, be partitioned by taxonomic aspects of model complexity and applicability, which are determined by the design and level of detail incorporated in the models of missile and aircraft flight dynamics, control systems, and sensor behavior. These taxonomic aspects could include the following mutually dependent considerations:

i) Nonlinear vs linear models of system dynamics.

ii) Inclusion vs exclusion of various lift, drag, and gravity terms.

iii) 3-dimensional kinematic behavior vs 2-dimensional kinematic behavior.

iv) Incorporation of constraints on missile and aircraft maneuvers.

v) Incorporation of deterministic and stochastic aspects of sensor and seeker dynamics, underlying constraints, and error sources.

We have classified the eight references cited in Section 2.1 based on these taxonomic considerations. The results of our classification are contained in Table 2-1. We emphasize that these eight references are merely

representative of recent work in missile evasion and allied studies, and that these references are, in their totality, neither complete with respect to the available literature, nor exhaustive with respect to its content. We also note that the design and rendering of a matrix or table for classifying the taxonomic aspects of a lengthy report, monograph, or proposal is subject to the interpretive analysis of the reviewer, and, hence, the reader should refer the original work to obtain the complete picture. It is also important to recognize that each report, monograph, or proposal was prepared with reference to some specific set of goals, and, hence, any blanket comparison between these works is to some degree inherently unfair.

Table 2-1a

Reference Identification and Miscellaneous Notes

Reports

Reference 1. (TASC, 1977):
Brown, C.M., and D.H. Johnson, "Fire Control Simulation (FICS) User's Manual," The Analytical Sciences Corp., TR-750-1, July 1977.

Reference 2. (Grumman, 1976):
Carpenter, G. and M. Falco, "Supercruise Vulnerability To Surface-To-Air Missile Threats," Grumman Aerospace Corp., 1976.

Reference 3. (Shinar & Steinberg, 1976):
Shinar, J. and D. Steinberg, "Analysis Of Optimal Evasive Maneuvers Based On A Linearized Two-Dimensional Kinematic Model," Technion - Israel Institute Of Technology, T.A.E. Report No. 230, 1976.

Reference 4. (Poulter, 1975):
Poulter, R.A., "Differential Game Guidance Versus Proportional Navigation For An Air-To-Air Missile," Air Force Institute Of Technology, GA/MC/75-5, December 1975.

Reference 5. (Grumman, 1975):
Carpenter, G. and M. Falco, "Analysis Of Aircraft Evasion Strategies In Air-To-Air Missile Effectiveness Models," Grumman Aerospace Corp., Report No. RE-506, August 1975.

Monographs

Reference 6. (Garnell & East):
Garnell, P. and D.J. East, Guided Weapon Control Systems, Pergamon Press, 1977.

Proposals - MLE Sensitivity Analysis

Reference 7. (FAAC, 1977):
FAAC Proposal FP402U/2762, "Part I - Technical Proposal For A Missile Launch Envelope Sensitivity Analysis," First Ann Arbor Corp., January 1977.

Reference 8. (Veda, 1977):
Veda Proposal 13005-77U/Q0105, "Technical Proposal Volume I - Missile Launch Envelope Sensitivity Analysis," Veda Corp., January 1977.

Continuation of Table 2-1a

Miscellaneous Notes

Reference 1, Abstract:

"This user's manual documents the FIre Control Simulation (FICS) computer program. FICS is a FORTRAN simulation designed for the evaluation of a wide range of defensive fire control processing techniques for a single missile-target in a three dimensional engagement. It contains realistic models for attacking missiles, aircraft maneuver dynamics, escorted aircraft, coarse tracking sensors, a fine tracker, and a weapon pointing and acquisition system. In addition, it contains algorithms for target tracking, threat and kill assessment, weapon handover, and flight control handover. ... It has full monte carlo statistical evaluation capabilities ... "

Reference 2, Abstract:

"This paper presents a new effectiveness methodology with which to quantitatively analyze vulnerability/survivability of aircraft to surface-to-air missile threats. Supercruiser survivability tradeoffs with maneuver performance and threat warning system parameters are presented for a known SAM threat. In addition the survivability sensitivity to a variety of penetration altitude and Mach number flight conditions has been examined.

The approach is a blend of optimal control theory, stochastic learning theory and dynamic simulation resulting in a learning algorithm which permits the evaluation of aircraft evasive maneuvering as an integral part of the survivability measure. The methodology develops optimal evasive strategies in the form of a feedback control policy based upon a discretized set of information states which are available as visual or threat warning system cues to the pilot. The algorithm develops evasive strategies using the optimization criteria of maximizing the survival probability for all possible missile launch conditions."

Reference 3, Abstract:

"Optimal evasion from proportionally guided missiles is analyzed assuming two-dimensional linearized kinematics. By this assumption a simple search technique can be used instead of the cumbersome solution of a two point boundary value problem. Due to the simplicity of this approach it is possible to include in the mathematical model factors which have been neglected in other analytic studies. It is demonstrated that these factors as: the exact dynamic structure of the guidance system, the location of the saturating element in the guidance loop, the limited roll rate of the evading aircraft, etc., have major effects on the optimal maneuver sequence and determine the order of magnitude of the resulting miss distance.

Comparison with studies, which used non-linear kinematic models, shows that the domains of validity of linearized kinematics and two-dimensional analysis coincide. In the case of optimal evasion assessment, both assumptions are limited in their validity to nearly head-on or tail chase engagements. To analyze engagements of other initial conditions a three-dimensional model is required. The method described in this paper can be extended for this type of three-dimensional study."

Continuation of Table 2-1a

Miscellaneous Notes

Reference 4, Abstract:

"Proportional navigation is a closed loop optimal control for the case of a linear dynamic model of the air-to-air missile intercept problem and a quadratic cost function (Ref ...).

This paper presents a differential game model of the intercept problem using nonlinear realistic dynamics, free final time and a terminal cost function related to probability of kill. With this model proportional navigation is no longer optimal and the extent of its nonoptimality is indicated for a range of saddle point solutions.

A guidance concept based on differential game theory is discussed and is compared to proportional navigation in an off line simulation. The considerable gains made by this scheme over proportional navigation provide the incentive to develop a real-time version."

Reference 5, Abstract:

"This report presents a new methodology with which to quantify missile effectiveness and aircraft vulnerability. The approach is a blend of applications of optimal control theory, stochastic learning theory, and simulation. This methodology permits an evaluation of aircraft evasive maneuvering and countermeasures deployment strategy as an integral part of the effectiveness/ vulnerability measurements. The strategy determination is a form of feedback control policy based upon a discretized set of information thresholds in the relative coordinate space as would be available to an evading aircraft pilot. The optimization criteria of an evading aircraft is that of maximizing the survival probability for all relative coordinates. The representative model chosen for illustration is evasion from a close range air-to-air IR guided missile."

Reference 6, Notes:

The subject matter of this monograph is based on lecture notes given to the Guided Weapon Systems (M.Sc.) Course at the English Royal Military College Of Science in Shrivenham, Swindon. At the time of the completion of this monograph (August 1976), this course was the only one of its type in the U.K. and had been given continuously for twenty six years.

The chapter headings are: The Performance Of Target Trackers; Missile Servos; Missile Control Methods; Aerodynamic Derivatives And Aerodynamic Transfer Functions; Missile Instruments; Autopilot Design; Line Of Sight Loops; Homing Heads And Associated Stability Problems; Proportional Navigation And Homing Guidance Loops; Wiener Filter Theory Applied To Guidance Loop Design; Modern Control Theory Applied To Guidance Loop Design; Kalman Filters.

Chapter 9, Proportional Navigation And Homing Guidance Loops, contains the bulk of the material in this monograph relating to the effects of target maneuver.

Continuation of Table 2-1a

Miscellaneous Notes

References 7 & 8, Notes:

The FAAC Technical Proposal For A Missile Launch Envelope Sensitivity Analysis contains a very concise and lucid description of the essential factors effecting the performance of an air-to-air missile of the AIM-9L variety. This proposal calls for the use of an extensive simulation model of the AIM-9L, which had been developed by FAAC, to generate the required MLE sensitivity information. Similarly, the Veda Technical Proposal For A Missile Launch Envelope Sensitivity Analysis calls for the use of an extensive six degree of freedom cross-coupled flyout simulation model for the AIM-9L. This latter model was available through the Naval Weapons Center (NWC) at China Lake.

These simulation models represented, as of January 1977, the most widely accepted digital computer simulation models of the AIM-9L. In light of the extensive detail incorporated in these AIM-9L models, the fidelity of the missile models used in References 7 & 8 will dominate the fidelity of the generic air-to-air missile models used in the other representative references.

Table 2-1b

MSL Dynamics

Reference(s)	Type #	Description
3, 6	I	Linearized 2-dimensional point mass model with load factor constraint.
5	II	Nonlinear 2-dimensional point mass model with load factor constraint.
1, 2, 4	III	Nonlinear 3-dimensional point mass model with load factor constraint.
7, 8	IV	AIM-9L industry/government standard.

MSL Autopilot-Airframe Dynamics

Reference(s)	Type #	Description
4	0	Zero order dynamics in roll and AOA.
5	I	Linear first order roll (or yaw) dynamics and zero order AOA dynamics.
1	II	Linear first order AOA dynamics and zero order roll dynamics.
2	III	Linear first order pitch and yaw dynamics.
3, 6	IV	Linear nth order dynamics for lateral acceleration with/without saturating elements.
7, 8	V	AIM-9L industry/government standard.

Stochastic Attributes

Reference(s)	Type #	Description
2 - 5, 7, 8	0	None - a purely deterministic model or application.
6	I	Angular noise and glint modules in monte carlo missile simulation.
1	II	Large scale monte carlo simulation model including effects such as: missile seeker/sensor noise and wind gust disturbances.

Continuation of Table 2-1b

Missile Seeker/Sensor Dynamics and Error Sources

Reference(s)	Type #	Description
2, 3, 4	0	None - no dynamical constraints or error sources are imposed in these models.
5	I	FOV limits, gimbal limits, gimbal rate limit, and first order seeker dynamics.
6	II	FOV limits, gimbal limits, gimbal rate limit, second order seeker dynamics, angular noise, and glint.
1	III	Noise corrupted measurements of target: range, azimuth, elevation, range rate, azimuth rate, elevation rate, and glint (optional).
7, 8	IV	AIM-9L industry/government standard.

Aircraft Dynamics

Reference(s)	Type #	Description
5	I	2-D kinematics with piecewise constant longitudinal acceleration (3 values: +a, 0, -a) - motion along simple arcs.
6	II	2-D kinematics with constant aircraft-missile closure rate and piecewise constant lateral acceleration.
3	III	Same as Type II with a ramp function modification of the piecewise constant acceleration profile when roll rate limits are assumed.
7, 8	IV	3-D kinematics with constant airspeed and zero order lateral acceleration dynamics with load factor constraint.
2, 4	V	Same as Type IV without constant airspeed.
1	VI	3-D kinematics with constant airspeed and first order roll and AOA dynamics with load factor constraint.

3.0 Heuristic Algorithms for Determining Evasion Strategies Suitable for Real-Time Onboard Computation

3.1 Introduction

In this chapter we examine the real-time computation of aircraft evasion strategies against a multiple missile threat using a game theoretic approach. We describe herein several related heuristic algorithms for determining evasion strategies suitable for real-time onboard computation. These algorithms were developed subject to the following goals and constraints. Each algorithm should:

i) require relatively minimal dynamic and parametric information,

ii) provide capability against an N - missile threat ($N \geq 2$),

iii) generate aerodynamically feasible aircraft maneuvers to meet aircraft design limitations and pilot stress limitations,

iv) be computable using foreseeable hardware, and

v) exhibit markovian behavior (be restartable from present state information).

3.2 Aircraft and Missile Models

For the purposes of this study we chose generic models of the F-4 aircraft (Phantom II - McDonnell Douglas) and AIM-9 missile (Sidewinder). Salient features of the aircraft and missile models employed in this study, include:

i) nonlinear models of system dynamics,

ii) the inclusion of various lift, drag, gravitational forces,

iii) 3-dimensional kinematic bevavior with point mass dynamics,

iv) the incorporation of constraints on missile and aircraft maneuvers, including: missile autopilot dynamics, aircraft roll rate limits, and missile/aircraft load factor constraints, and

v) the assumption of a deterministic environment and perfect tracking.

The choice of this type of model was made to achieve a balance between simulation fidelity and computational complexity, in the context of taking a first cut at the real-time computation of aircraft evasion strategies against a multiple missile threat. The attributes of the aircraft and missile models incorporated in this study are closely matched by the dynamical models used in (Poulter, 1975). The differences include the addition of missile autopilot dynamics and aircraft roll rate limitations in the present study.

The details of the F-4 aircraft and AIM-9 missile dynamics used in the present study are contained in Table 3-1.

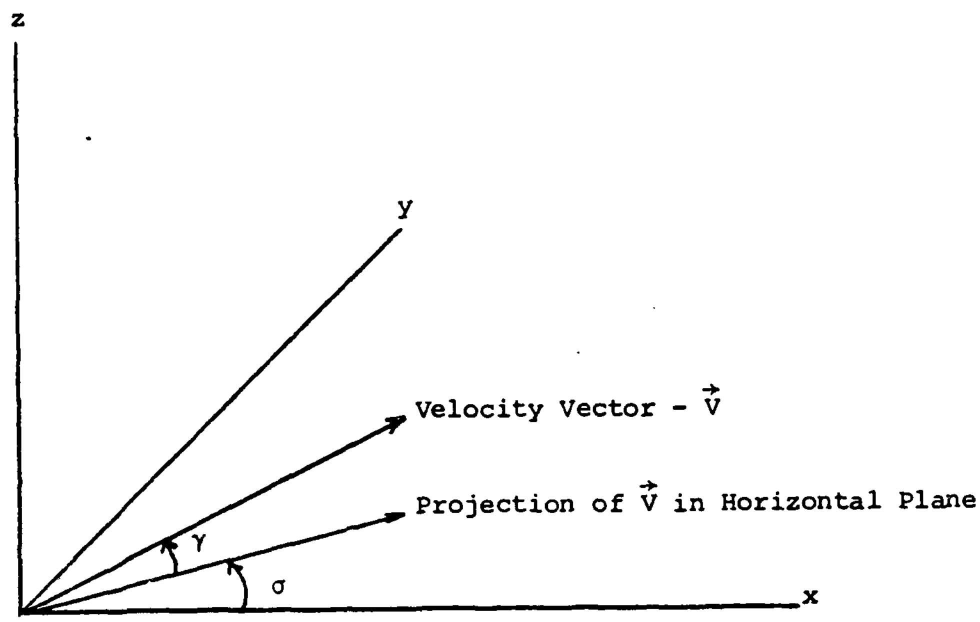
Table 3-1a

Definition of Dynamic Variables and Parametric Values which appear in the Aircraft and Missile Models

Generic Variables and Parameters Common to both Aircraft and Missile Models:

x, y, z	-	earth based inertial coordinate system (Cartesian),
v	-	airspeed (magnitude),
γ	-	angle defined by velocity vector and horizontal (x-y) plane,
σ	-	angle defined by the projection of the velocity vector in the horizontal plane and the x-axis,
T	-	thrust along vehicle center line,
α	-	angle of attack (AOA),
μ	-	vehicle bank angle,
L	-	lift force (normal to velocity vector and wing planform),
D	-	drag force (colinear with velocity vector),
m	-	vehicle mass,
g	-	gravitational constant,
n	-	vehicle load factor ($n = L/mg$),
C_L	-	coefficient of lift,
C_D	-	coefficient of drag,
S	-	characteristic area,
ρ	-	air density ($\rho = \rho_o e^{-\beta z}$)

Figure 3-1a



Earth Based Coordinate System

Table 3-1b

F-4 Aircraft Dynamics

$dx/dt = v\cos(\gamma)\cos(\sigma)$

$dy/dt = v\cos(\gamma)\sin(\sigma)$

$dz/dt = v\sin(\gamma)$

$dv/dt = (T\cos(\alpha)-D)/m \;-\; g\sin(\gamma)$

$d\gamma/dt = (L+T\sin(\alpha))\cos(\mu)/mv \;-\; g\cos(\gamma)/v$

$d\sigma/dt = (L+T\sin(\alpha))\sin(\mu)/mv\cos(\gamma)$

$T = k_1 + k_2 z + k_3 v$

$L = C_L \rho v^2 S/2$

$C_L = C_{L_\alpha} \alpha$

$D = C_D \rho v^2 S/2$

$C_D = C_{D_o} + \kappa C_L^2$

Parametric Values

m	=	1243.0 slugs
k_1	=	22347.0 lbf (Afterburner Off); 35347.0 lbf (Afterburner On)
k_2	=	- 0.7018 lbf/ft
k_3	=	18.141 lbf-sec/ft
C_{L_α}	=	3.8986 rad^{-1}
C_{D_o}	=	0.01675
κ	=	0.223
S	=	530.0 ft^2

Conti..uation of Table 3-1b

Constraints on Aircraft Dynamics

The maximum allowed load factor is 8.0 g. The angle of attack is limited to a maximum value of 25.0 degrees. This corresponds to approximately 29 AOA Units on the cockpit AOA Indicator, based on the relation,

$$\text{AOA Units} = 1.03(\alpha(\text{degrees}) + 3.3)^{\dagger}.$$

The maximum roll rate is limited to 600 deg/sec. This is three times the maximum suggested roll rate in the F-4 "dash - one manual" (USAF, 1977). The consequences of rolling at this higher rate could include damage to missile stores. However, we have learned through informal conversations with F-4 pilots, that F-4 roll rates in excess of 600 deg/sec could be employed in certain combat situations.

† McDonnell Douglas Co. - Private Communication

Table 3-1c

AIM-9 Missile Dynamics

$dx/dt = v\cos(\gamma)\cos(\sigma)$

$dy/dt = v\cos(\gamma)\sin(\sigma)$

$dz/dt = v\sin(\gamma)$

$dv/dt = -(D/m + g\sin(\gamma))$

$d\gamma/dt = L\cos(\mu)/mv \quad - \quad g\cos(\gamma)/v$

$d\sigma/dt = L\sin(\mu)/mv\cos(\gamma)$

$d\alpha/dt = -(\alpha - \alpha_{com})/\tau;\ \alpha_{com}$ = AOA commanded by missile guidance system

$L \quad = C_L \rho v^2 S/2$

$C_L \quad = C_{L_\alpha} \alpha$

$D \quad = C_D \rho v^2 S/2$

$C_D \quad = C_{D_o} + \kappa C_L^2$

Parametric Values

m	=	3.2 slugs
C_{L_α}	=	22.918 rad^{-1}
C_{D_o}	=	0.7
κ	=	0.042
S	=	0.223 ft^2
τ	=	0.15 sec

Continuation of Table 3-1c

Missile Guidance Law

Proportional Navigation Relations:

Let θ and ψ denote respectively, the yaw and pitch line of sight (los) angles, for the los defined by the missile and aircraft depicted in Figure 3-1b. Let $d\theta/dt$ and $d\psi/dt$ denote their respective rates.

Classical proportional navigation (AFA, 1975) provides that the missile pitch and yaw rates ($d\gamma/dt$ & $d\sigma/dt$) are determined by: $d\gamma/dt = R_p d\psi/dt$, and $d\sigma/dt = R_y d\theta/dt$, where R_p and R_y are the Proportional Navigation Constants in pitch and yaw. Hence,

$$\tan(\mu) = (R_y d\theta/dt)/(R_p d\psi/dt + g\cos(\gamma)/v),$$

$$n = (R_y d\theta/dt) v\cos(\gamma)/\sin(\mu), \text{ if } d\theta/dt \neq 0$$

and

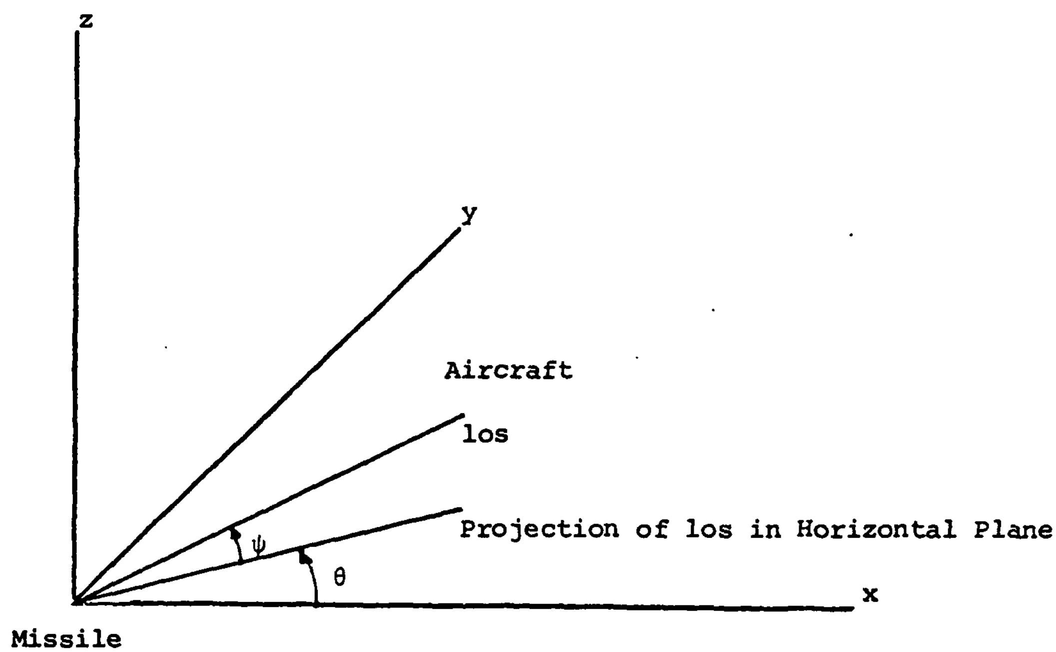
$$n = (vR_p d\psi/dt + g\cos(\gamma))/g\cos(\mu), \text{ if } d\theta/dt = 0.$$

In this simulation we have specified that $R_p = R_y = 4.5$. We remark that the AIM-9L is roll rate stabilized, and, hence, the pitch and yaw commands in the present simulation are unrealistic to the extent that the lateral acceleration components will vary by as much as a factor of $(2)^{\frac{1}{2}}$, and the instantaneous value of this factor will depend on the roll history of the missile (FAAC, 1977).

Constraints on Missile Dynamics

The missile load factor is constrained to be less than or equal to 15.0 g in absolute value. No roll rate limit is specified. The AOA rate is limited through the first order dynamics specified previously in this Table ($\tau = 0.15$ sec).

Figure 3-1b



Definition of Missile-Aircraft los Angles

3.3 Remarks on Modeling Philosophy and Underlying Assumptions

The choices of aircraft and missile models incorporated in this study were made to capture those attributes of both dynamical systems which were felt to be important in determining significant aspects of both global and local characteristics of missile-aircraft interception behavior. In comparison with the missile models employed in the eight exemplary references, which were reviewed in Chapter 2, the missile model employed in the present study is basically similar in detail to the deterministic aspects of the generic missile model described in (TASC, 1977), the generic missile model described in (Grumman, 1976), and the generic missile model described in (Poulter, 1975) - augmented with autopilot dynamics. In making a similar comparison among the aircraft models, we note that the aircraft model employed in the present study is similar in detail to those described in (Grumman, 1976) and (Poulter, 1975) - when both are augmented to include aircraft roll rate limits.

3.3.1 Attributes of Aircraft-Missile Models in the Context of Evasive Aircraft Maneuvering

The choice of 3-dimensional nonlinear models including lift, drag, and gravitational forces, as well as the incorporation of constraints such as missile autopilot dynamics, aircraft roll rate limits, and missile-aircraft load factor limits, provides the means for accounting for significant aspects of the global characteristics of missile-aircraft interception behavior.

The assumption of "linearizability" has been invoked in prior studies, e.g., (Shinar & Steinberg, 1976) and (Garnell & East, 1977). This assumption, when coupled with a deterministic model, allows one to analyze the evasion problem as a <u>fixed terminal time</u> problem with vehicles that have constant airspeed. The validity of this type of analysis is limited basically to end game analyses on the order of ten missile autopilot-airframe time constants

in duration, with an engagement geometry which is within a small perturbation from tail-chase or head-chase conditions (Shinar & Steinberg, 1976), or within a small perturbation from a collision - constant velocity - flight path (Garnell & East, 1977). The analysis based on linearization is useful in the end game, when the conditions for its validity are met. However, the linearization approach does not allow for the effect of global maneuvers which are designed to evade missiles by initiating maneuvers outside the end game envelope, or to provide for entry into a "favorable set" of end game states. Simulation results, obtained in the present study, indicate that relative missile-aircraft load factor capability, missile autopilot-airframe time constants, aircraft roll rate limit, engagement geometry, relative airspeed, and maneuver timing play a significant role in the outcome of an evasive maneuver. The factors associated with engagement geometry and airspeed variability are global in extent, and therefore, these factors are not amenable to a linearized analysis outside the end game envelope.

It is important to note that the timing of a maneuver outside the end game envelope can significantly affect the actual time at which an intercept can occur. Hence, the goal of maximizing the missile-aircraft miss distance at an a priori specified terminal time can be quite misleading, because the miss distance at the a priori terminal time can be quite large, but, without any further maneuvering by the aircraft, the missile still may achieve an intercept at a later time.

The importance of these previous considerations have been noted in earlier studies, e.g., see the exemplary references cited in Chapter 2. The contribution of the present study focuses on the evasive maneuver strategy determination in a multiple missile environment, when the luxury of a single or one at a time end game analysis may not be available.

3.3.2 Potentially Important Factors Omitted in the Missile Model

The major limitation of the missile model in the present study is the unrealistic sensor-seeker characterization. Factors such as: seeker limits, seeker dynamics, stochastic effects, and blind range are ignored entirely. This lack of realism in seeker behavior makes the missile system appear to be much more effective than it actually is. This and other modeling considerations are the subject of a further study of evasive tactics in a multiple missile environment. Finally, the important issue of counter measures has also been ignored. From a systems point of view, it is clear that the potential use of counter measures should be considered in a missile evasion study. However, for practical reasons, it was felt to be outside the potential scope of this present study.

3.4 Aircraft Evasion Strategies - Basic Considerations

3.4.1 Qualitative Aspects

In the analysis and synthesis of evasive maneuver strategies, it is useful to partition the problem into two phases. Phase I will be referred to as the Extra-End-Game (EEG) phase, and will denote that portion of the missile-aircraft engagement outside the end game envelope. Phase II will be referred to as the Intra-End-Game (IEG) phase, and will denote that portion of the missile-aircraft engagement inside the end game envelope. It is important to recognize that the end game envelope is not a totally precise notion. Qualitatively, we shall mean by the end game envelope that portion of the missile-aircraft engagement in which the missile is within 10 - 15 missile time constants of an interception or point of closest approach (FAAC, 1977). It is evident that the state space description of the end game envelope will depend on the aircraft maneuvers in the EEG phase of the engagement. This coupling between the end game envelope and the EEG phase maneuvering of the aircraft

is conceptually quite similar to the variations that exist between MLE's in the cases referred to as "target aware" versus "target unaware" conditions.

During the EEG phase, aircraft maneuvers initiated against a missile launched inside the target unaware MLE would generally cause the missile to maneuver in response to the changing orientation of the missile-aircraft los vector. With significant "time-to-go" available (relative to the missile autopilot-airframe time constants) the missile would be expected to "follow" the aircraft's maneuvers. Hence, the goal of the aircraft would be to seek to maneuver in such a way as to:

i) cause the missile to give up enough energy so as to "place the missile beyond its maximum range" (FAAC, 1977), or

ii) cause the missile and aircraft to enter the end game envelope in a region of state space favorable to the aircraft, i.e., a region from which the aircraft could outmaneuver the missile based on considerations such as: missile autopilot-airframe time constants, relative turning radii, and seeker behavior.

It is important to note that, depending on initial conditions, there may be "blends" of strategies based on i) and ii) which are reasonable for the aircraft to use. In addition to these considerations, the multiple missile threat adds to the complexity of the decision environment by constraining the aircraft to avoid situations where an escape from one missile leads to a "setup" for a second missile.

3.4.2 Quantitative Methods

Earlier studies, e.g., (Poulter, 1975), (Shinar & Steinberg, 1976), and (Garnell & East, 1977) consider the application of <u>optimal control techniques</u> to determine aircraft evasion strategies. The work by (Poulter, 1975) addresses the determination of optimal maneuvers in the EEG phase of the problem. The computational aspects of the problem are fraught with difficulties which are typical of free terminal time nonlinear programming problems. The compu-

tational problem may exhibit local maxima and, hence, the determination of a global solution may be difficult to obtain, and, in any case, very time-consuming to achieve. In the context of the overall goals of this present study, there seems to be no reasonable possibility of attaining a real-time global solution to the optimal maneuver problem (against even a single missile) starting from the EEG phase.

If one seeks to optimize the maneuver starting from the IEG phase, by assuming that the terminal time is known a priori and that linearization is valid, one is led to a set of computationally tractable approximations, see for example, (Shinar & Steinberg, 1976) and (Garnell & East, 1977).

The focus of the present study is on the real-time determination of evasive maneuvers in a multiple missile environment, where the luxury of a single or one at a time end game analysis may not be available, i.e., the analysis and implementation of evasive maneuvering must begin in the EEG phase. This brings us to the introduction of the heuristic algorithms which have been developed for this purpose.

3.4.3 Geometric Aspects of Missile Guidance

Before we describe the heuristic algorithms, it is useful to review two well-known concepts in missile guidance "lore",

i) the geometric-kinematic concept known variously as the Intercept, Maneuver, or Guidance Plane (FAAC, 1977) and (Garnell & East, 1977), and

ii) the geometric-kinematic concept known as Anti-Proportional Navigation (TASC, 1977).

The Maneuver Plane:

The maneuver plane is defined to be that plane determined by the missile-aircraft los vector and the aircraft velocity vector.

"For a non-maneuvering target, the optimum (missile) heading angle lies in the plane defined by the target (aircraft) velocity vector and the los at launch. This plane is sometimes called the intercept or maneuver plane. It

is an estimate of this optimum heading that the fire control system usually computes and provides to the pilot for steering cues. It is this same optimum heading from which heading errors are measured. Since launches from other than optimum heading require the missile to maneuver more than it would if launched from the optimum, more missile energy is consumed and performance is affected." (FAAC, 1977; p. 20).

When the missile velocity vector lies in the maneuver plane (as defined above), the maneuver plane can be defined equivalently as that plane determined by the missile-aircraft los vector and the relative velocity vector of the missile and aircraft. We will employ this definition in the sequel.

Anti-Proportional Navigation:

The concept of anti-proportional navigation can be simply stated as that aircraft maneuver which instantaneously maximizes the missile-aircraft line of sight rate (TASC, 1977).

Let a_ψ and a_θ denote respectively the aircraft acceleration in pitch and yaw, as measured in missile-aircraft los coordinates (see Figure 3-2). Then, a simple exercise in calculus demonstrates that, the missile-aircraft los rate is maximized instantaneously (myopic maximization) when the aircraft acceleration vector has maximum magnitude and an orientation defined by:

$$(a_\psi/a_\theta) = (d\psi/dt)/(d\theta/dt)\cos(\psi),$$

$$a_r = 0,$$

and the signs of the components are chosen to increase the instantaneous los rate.

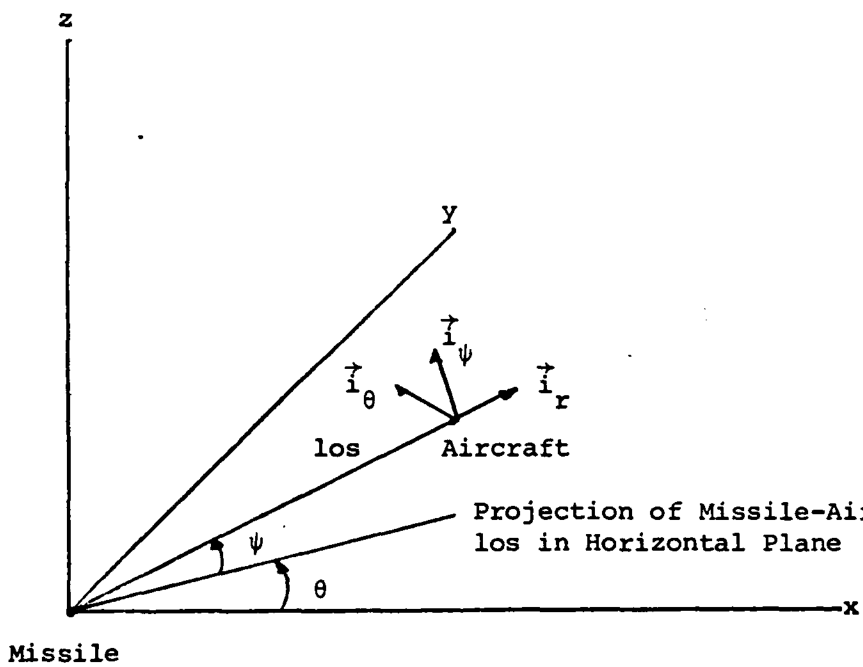
The Relationship Between Anti-Proportional Navigation and the Maneuver Plane:

The vector cross product $\vec{G} \triangleq \vec{los} \times \vec{v}_{rel}$ defines a normal vector to the maneuver plane, which is expressible in missile-aircraft los coordinates by:

$$\vec{G} = (r d\psi/dt)\vec{i}_\theta + (-r\cos(\psi)\, d\theta/dt)\vec{i}_\psi,$$

where $\vec{i}_\psi$ and $\vec{i}_\theta$ denote the unit vectors (in los coordinates) in the pitch and yaw directions respectively. We observe that the ratio of the components

Figure 3-2



Definition of Missile-Aircraft los Unit Vectors

(a_ψ/a_θ) of $\vec{G}$ is the negative reciprocal of the ratio (a_ψ/a_θ) formed by the components of the anti-proportional acceleration vector defined previously. Hence, the anti-proportional acceleration vector lies in the maneuver plane.

3.5 Heuristic Algorithms

Herein we describe four heuristic algorithms for countering a multiple missile threat. The algorithms are defined for the case of N = 2 missiles; however, the respective extensions to the general case (N > 2) is essentially trivial in each instance. To simplify the following discussion, we denote the algorithms by: Algorithm 1, .., Algorithm 4.

Let $\vec{G}_i$ (i=1,2) denote the unit normal vector associated with the maneuver plane of missile i. Let $\vec{\Gamma}_i$ (i=1,2) denote the unit vector associated with the commanded (desired) direction defined by the anti-proportional navigation rule associated with missile i. We note that in los coordinates, the ratio of pitch to yaw components of $\vec{G}_i$ is the negative reciprocal of the ratio of pitch to yaw components of $\vec{\Gamma}_i$, i = 1, 2. Algorithms 1 & 2 are defined in terms of the maneuver plane unit normals $\vec{G}_1$ & $\vec{G}_2$; whereas, Algorithms 3 & 4 are defined in terms of the anti-proportional navigation unit vectors $\vec{\Gamma}_1$ & $\vec{\Gamma}_2$. Finally, let $\vec{A}(\mu)$ denote the instantaneous total acceleration vector of the aircraft. The orientation of the aircraft total acceleration vector is a function of the aircraft bank angle μ, as well as the angle of attack and the velocity vector. The magnitude of $\vec{A}(\mu)$ depends on the instantaneous aircraft drag, lift, and thrust forces, as well as the force of gravity.

Algorithm 1:

Define $J_i(\mu)$ by,

$$J_i(\mu) = \vec{G}_i \cdot \vec{A}(\mu), \quad i = 1, 2.$$

Define $P(\mu,\lambda)$ by,

$$P(\mu,\lambda) = |\lambda J_1(\mu) + (1-\lambda) J_2(\mu)|,$$

where $-\pi \leq \mu \leq \pi$ rad, and $0 \leq \lambda \leq 1$.

Algorithm 1 dictates that at each decision instant a pair (μ^*,λ^*) be determined such that:

$$P(\mu^*,\lambda^*) = \min_{\lambda} \max_{\mu} P(\mu,\lambda) = \max_{\mu} \min_{\lambda} P(\mu,\lambda),$$

until a closure rate negative (CRN) condition is achieved for one of the missiles. Then, μ^* is determined by:

$$\mu^* = \arg\max |J_i(\mu)|,$$

where the i^{th} missile is assumed to be in a closure rate positive (CRP) condition.

Here we assume that a saddle point pair (μ^*,λ^*) for $P(\mu,\lambda)$ exists and refer the reader to Chapter 5 of this report to review relevant related saddle point existence results.

The heuristic of Algorithm 1 requires, in effect, the myopic determination of an instantaneous saddle point pair. This saddle point has the following interpretation: The instantaneous bank angle μ^* is precisely that bank angle which maximizes the minimum value of the absolute value of the projection of the aircraft acceleration vector on a "synthetic" maneuver plane normal $\vec{G}(\lambda)$, obtained by forming a linear combination (convex combination) of the individual maneuver plane normals:

$$\vec{G}(\lambda) = \lambda\vec{G}_1 + (1-\lambda)\vec{G}_2,$$

i.e.,

$$P(\mu,\lambda) = |\vec{G}(\lambda)\cdot\vec{A}(\mu)|.$$

We note that

$$P(\mu,1) = |\vec{G}_1\cdot\vec{A}(\mu)|, \text{ and}$$

$$P(\mu,0) = |\vec{G}_2\cdot\vec{A}(\mu)|.$$

Thus, the value of μ which maximizes $P(\mu,1)$, is that value of μ which instantaneously maximizes the absolute value of the projection of $\vec{A}(\mu)$ on $\vec{G}_1$, i.e., that value of μ which maximizes the aircraft acceleration normal to the

maneuver plane associated with missile 1.

The heuristic of Algorithm 1 provides a degree of weighting or importance λ^* and $(1-\lambda^*)$ to be associated with the maneuver plane of the respective missile threats. As the weighting factor λ^* varies from one to zero in value, the weight associated with the maneuver plane of missile 1, λ^*, is decreased, and the weight associated with the maneuver plane of missile 2, $(1-\lambda^*)$, is increased accordingly.

In summary, the heuristic of Algorithm 1 allows the amount of aircraft acceleration which is available at a given decision instant to be apportioned in such a way that a balance is achieved in "attempting" to simultaneously maximize the aircraft acceleration normal to each maneuver plane.

Algorithm 2:

Based on the notation introduced in the presentation of Algorithm 1, Algorithm 2 dictates that at each decision instant μ^* be determined by:

$$\mu^* = \arg\max\ (\min(|J_1(\mu)|,|J_2(\mu)|)),$$

until a CRN condition is achieved for one of the missiles. Then, μ^* is determined by:

$$\mu^* = \arg\max\ |J_i(\mu)|,$$

where the i^{th} missile is assumed to be in a CRP condition.

The heuristic of Algorithm 2 requires, in effect, the myopic determination of a bank angle μ^* which maximizes the minimum (over all missiles) of the absolute value of the projection of the instantaneous aircraft acceleration on the normal vector to the missile maneuver plane. This is equivalent to the following max min problem:

$$\max_{\mu}\ \min_{\lambda}\ (\lambda|J_1(\mu)| + (1-\lambda)|J_2(\mu)|),$$

where $-\pi \leq \mu \leq \pi$ rad, and $\lambda \in \{0,1\}$.

The heuristic of Algorithm 2 differs from that of Algorithm 1 in that

Algorithm 2 is "binary" and allows no smooth blending of objectives by the assignment of an intermediate degree of importance or weight to each missile threat in the objective function.

In summary, under Algorithm 2, the full attention of the aircraft is captured at each decision instant by just one of the two missiles when both missiles are closing on the aircraft.

Algorithm 3:

Algorithm 3 is the dual of Algorithm 1 with $\vec{G}_1$ replaced by $\vec{\Gamma}_1$ and $\vec{G}_2$ replaced by $\vec{\Gamma}_2$. Here, the objective is to seek a trade-off between the line of sight rate associated with each missile, as opposed to the aircraft acceleration normal to the individual maneuver planes.

Algorithm 4:

Algorithm 4 is the dual of Algorithm 2 with $\vec{G}_1$ replaced by $\vec{\Gamma}_1$ and $\vec{G}_2$ replaced by $\vec{\Gamma}_2$. The interpretation of the heuristic associated with Algorithm 4 is similar to that of Algorithm 2, with the objective now being line of sight rate maximization, as opposed to "maximizing" aircraft acceleration normal to the individual maneuver planes.

3.6 Information Requirements

One of the goals of this present study was the development of a real-time algorithm with relatively minimal requirements in terms of dynamic data and the description of missile system dynamics. It is important to note that all four heuristic algorithms described in this study require only information delineating: the relative velocity between each missile and the aircraft, and each missile-aircraft los. No a priori knowledge of the missiles' dynamics or control systems is required. This is not to say, however, that additional information would be useless, since a more complete description of each missile might permit an approximate faster than real-time flyout simulation to be run in parallel with the heuristic evasive maneuver decision model to enhance the decision making process.

4.0 Computer Simulation Models for Determining and Evaluating the Global and Local Characteristics of Evasive Maneuvers

4.1 Introduction

In this chapter we describe the details of the simulation models based on the four heuristic algorithms delineated in Chapter 3, and we present and analyze the simulation results pertaining to the exercising of these four algorithms against seven representative missile-aircraft engagement scenarios. The related FORTRAN IV source programs are contained in Appendix A of this report.

4.2 The Simulation Paradigm and Engagement Scenarios

4.2.1 The Simulation Paradigm

The four simulation models developed for use in this study are flyout simulations for evaluating and comparing the four heuristic evasion algorithms defined in Chapter 3. The individual programs are referred to as ACDYN.91, ..., ACDYN.94, and correspond respectively to Algorithms 1 - 4. The programs ACDYN.91 - ACDYN.94 have many features in common. Hence, program ACDYN.91 will be described in detail, and the relevant differences which characterize 92 - 94 will be provided to complete the program package. The source program for ACDYN.91 is complete with the exception of certain nonessential library graphics routines for producing plots. Flowcharts for programs 91 & 92 are included in this chapter. Programs 91 & 93 are virtually identical -- the sole exception pertains to the remarks in Section 3.5 (p. 3-20) in reference to the vectors $\vec{G}_i$ and $\vec{\Gamma}_i$, $i = 1, 2$. This same statement holds as well for programs 92 & 94.

The flyout simulations are based on the dynamical models of the F-4 aircraft and AIM-9 missile which are delineated in Chapter 3. All missile-aircraft engagement scenarios considered in this study exhibit the following salient characteristics:

i) The missiles are assumed to be coasting (thrust equals zero) when the evasive tactics are initiated.

ii) Without evasive maneuvering, all missile-aircraft engagements would result in a first missile kill within 4 - 8 seconds from the opening of the engagement scenarios.

iii) The integration time step is defined by

STEP = min(T,0.5*DSV/VREL),

where: DSV denotes the current separation distance between the closest missile and the aircraft; VREL denotes the magnitude of the relative velocity between the closest missile and the aircraft; and T equals 0.1 seconds.

iv) The aircraft begins the maneuver process dictated by the relevant algorithm at the time instant t = 0.1 sec after the opening of the engagement scenario. Hence, a detailed study of maneuver initiation timing was not carried out as a part of this present study.

v) During the maneuver process the aircraft afterburner is on, and the aircraft load factor is 8 g, or the aircraft is limited to a maximum AOA of 25 degrees.

vi) The aircraft is roll rate limited to 600 deg/sec. Hence, if the commanded bank angle based on the decision algorithm is greater than 600.*STEP degrees from the present roll position, the commanded roll increment is limited to 600.*STEP.

Flowcharts for ACDYN.91 & ACDYN.92

The flowcharts for programs ACDYN.91 and ACDYN.92 are included in Figures 4-1(a - h). Figure 4-1a depicts the overall organization of the generic ACDYN simulation program. The first two subroutines, INIT and VALUE, define the heart of the program (refer to Figures 4-1(a - f)). Subroutine INIT (Figure 4-1b) allows the user to either interactively initialize an engagement scenario, or read and update a file with such information. Subroutine VALUE (Figures 4-1(c - f)) provides the necessary flyout simulation logic and report generation. The variations between programs ACDYN.91 - 94 occur in subroutine VALUE, specifically, within the program module for determining the values of μ^* and λ^* (see Figures 4-1(c, e, & f)). Figures 4-1(e & f) flowchart the determination of the pair (μ^*,λ^*) within ACDYN.91 & 92 respectively. The generic functional notation $J(\mu,\lambda)$, which appears in the caption of Figures 4-1(e & f),

refers to the relevant game theoretic kernel functions defined explicitly in heuristic Algorithms 1 & 2 (Section 3.5, pp. 3-17 & 3-19). The remainder of the program, subroutines INTBOX & PLOUT (Figures 4-1(g & h)), provide the necessary integration capability and the calls to the library plotting routines.

4.2.2 The Engagement Scenarios

The performance of the heuristic algorithms was studied by means of comparative scenario analysis. The four heuristic algorithms were exercised against seven representative missile-aircraft engagement scenarios. The initial conditions at the opening of each scenario are summarized in Table 4-1. The x-y position and horizontal velocity of the missiles and aircraft in each scenario are depicted vectorially in Figures 4-2(a - c). The engagements were chosen to depict a combination of missile threats which ranged through mixtures of tail-chase, head-chase, and off-beam launch conditions. The engagements included scenarios (1 - 3) where one missile is above the aircraft initially and the second missile is launched from below, as well as coplanar initial conditions (5 & 7), and cases (4 & 6) where both missile are launched from below the aircraft. The engagements all lead to multiple hits if no aircraft maneuvering occurs.

4.3 Algorithm Performance

Summary and Ranking

The performance of Algorithms 1 - 4 parameterized by scenario and individual missile identification is summarized in Table 4-2a. The detailed results of each of nine scenario/algorithm pairs in which the aircraft achieved multiple misses are contained in Tables 4-3 through 4-11, and Figures 4-3 through 4-11. The definitions of program variables which appear in Tables 4-3 through 4-11 are contained in Table 4-2b.

In order to analyze and compare the performance (Table 4-2a) of the four

heuristic algorithms against the seven multiple missile threat scenarios defined in Table 4-1, it is necessary to adopt a suitable performance measure. A game theoretic approach suggests that the minimum of the miss distance, associated with missiles 1 & 2, be adopted as a measure of performance for each algorithm-scenario pair. Based on this approach, we observe that there is no single algorithm whose performance dominates the remaining three algorithms on a scenario by scenario basis. If, in addition, we calculate the number of multiple misses, we observe that the algorithms in decreasing order of performance are: Algorithms 1, 4, 2, & 3.

Detailed Simulation Output

Tables 4-3 through 4-11 contain the initial and final state information associated with the nine successful (multiple miss) algorithm-scenario pairs contained in Table 4-2a. In addition, these nine tables contain the minimum miss distance information collected as part of the overall simulation report generation. Figures 4-3 through 4-11 are multiple figures, e.g., Figure 4-3 contains Figures 4-3a, ..., 4-3e. Figures associated with Algorithm 1 contain five graphs (a - e) which depict: (a) Aircraft Bank Angle μ^* vs time; (b) λ^* x 100 vs time; (c) the Projection of the Aircraft and Missile Trajectories in the (x-y) Plane; (d) the Projection of the Aircraft and Missile Trajectories in the (z-x) Plane; (e) the Projection of the Aircraft and Missile Trajectories in the (z-y) Plane. Figures associated with Algorithms 2 & 4 contains four graphs (a - d) which depict: (a) Aircraft Bank Angle vs time and λ^* vs time (the symbol "F" denotes λ^*, where an "F - value" of 50 denotes $\lambda^* = 1$, and an "F - value" of -50 denotes $\lambda^* = 0$); (b) - (d) depict respectively the projections of the aircraft and missile trajectories in the (x-y), (z-x), and (z-y) planes.

Figure 4-1a

ACDYN Flowchart

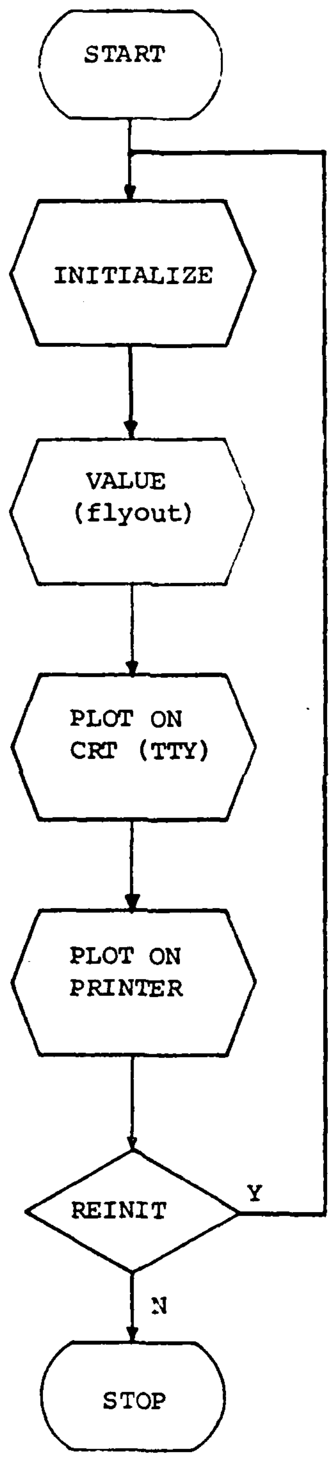


Figure 4-1b

Subroutine INIT

(Initialization)

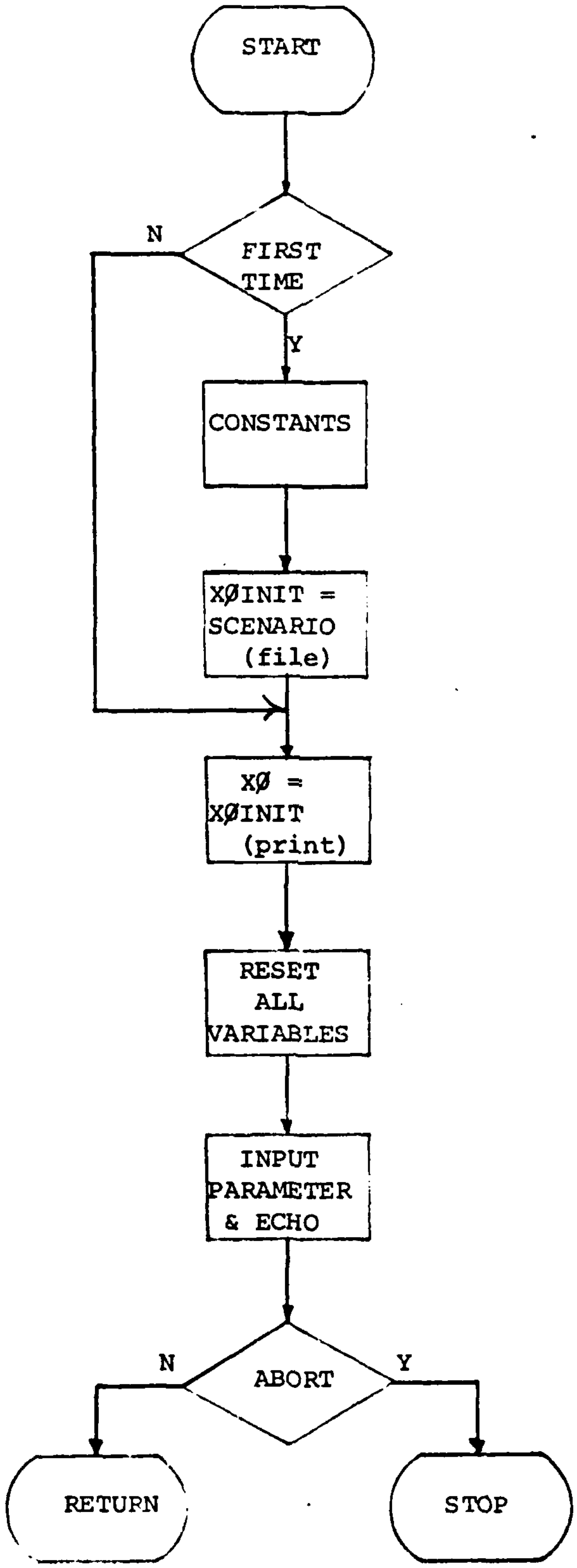


Figure 4-1c

Subroutine VALUE

(flyout simulation)

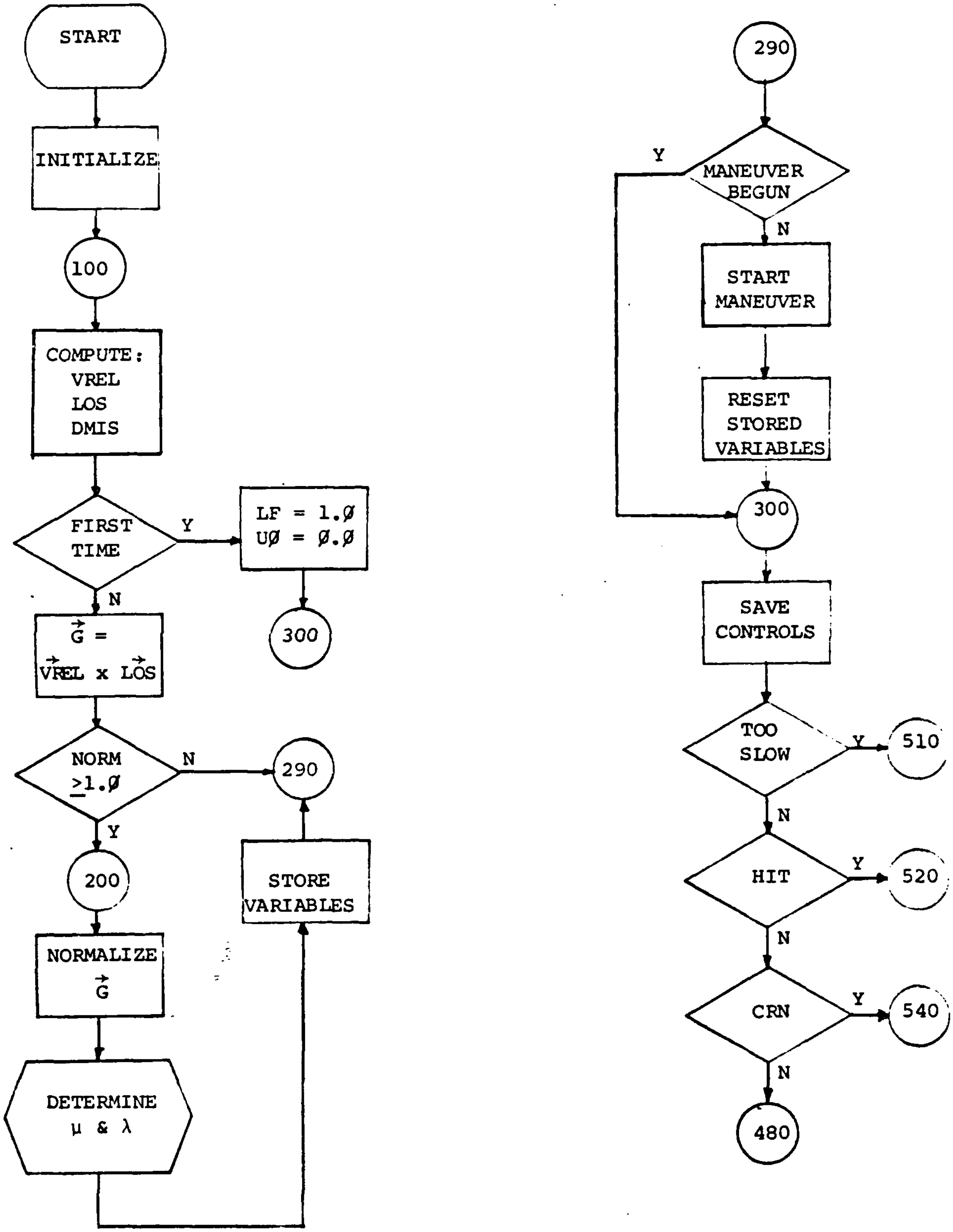


Figure 4-1d
Continuation of
Subroutine VALUE

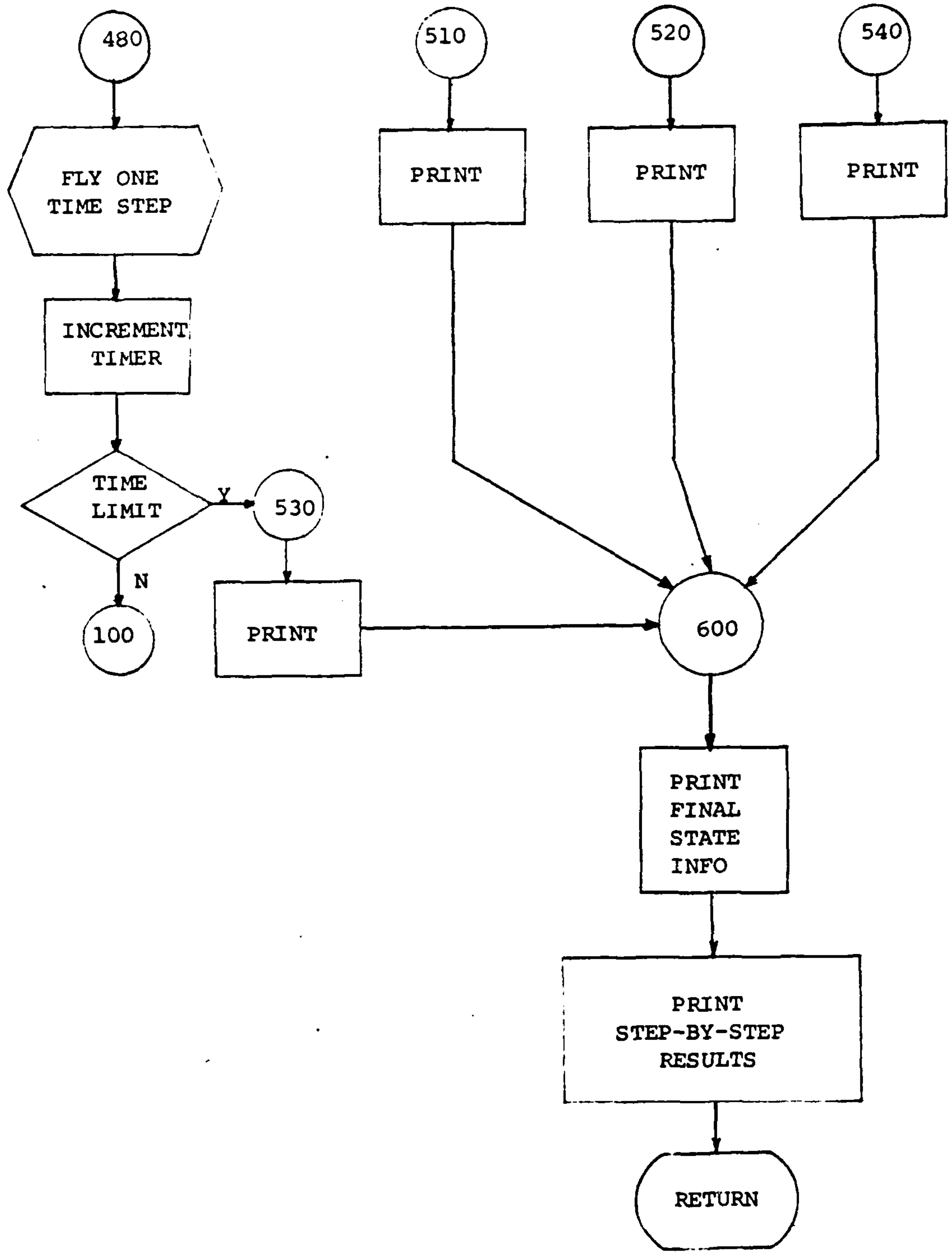


Figure 4-1e

Program Module within Subroutine VALUE to determine λ & μ via:

$\min_{\lambda} \max_{\mu} J(\mu,\lambda)$ ACDYN.91 (Algorithm 1)

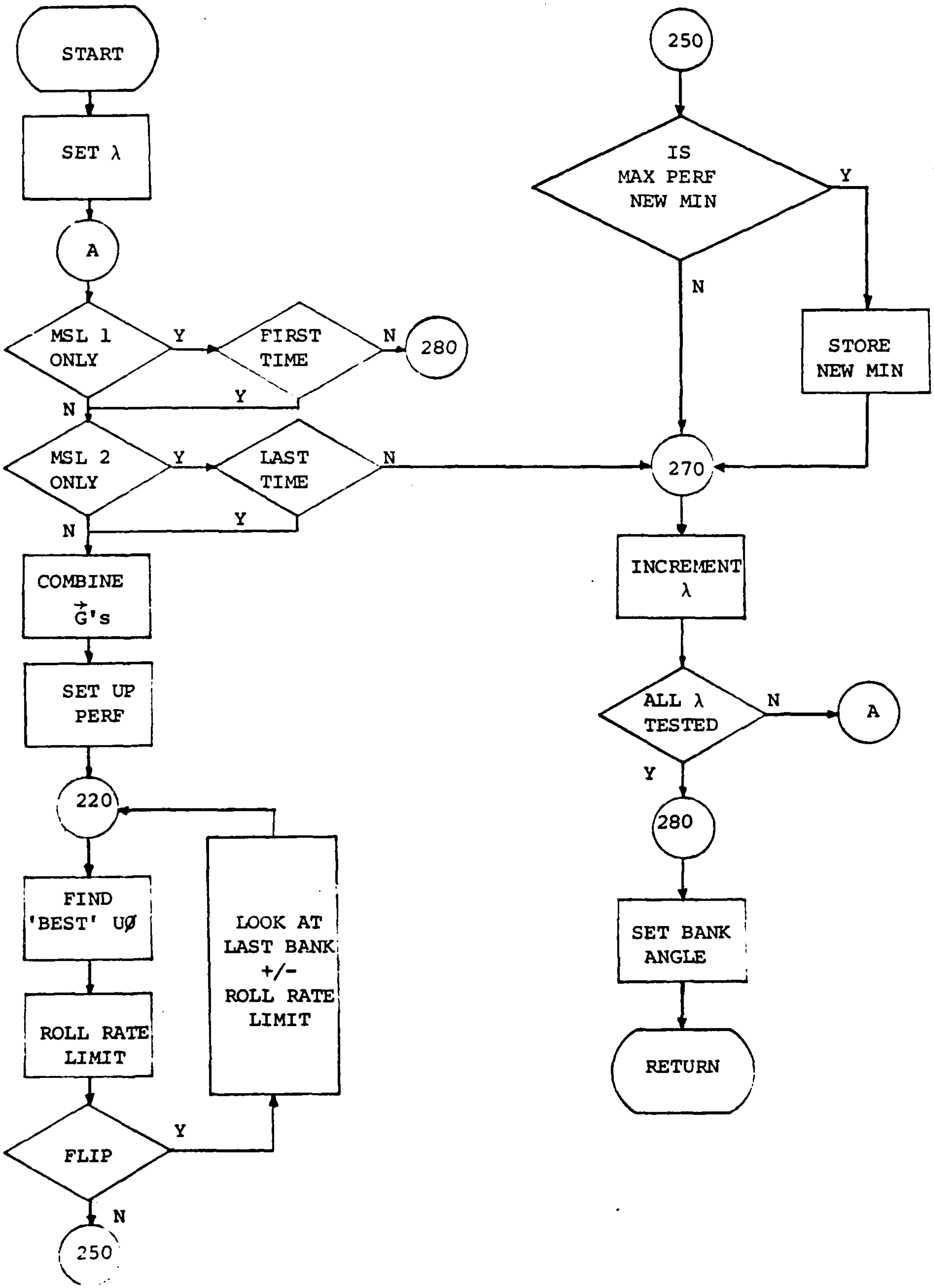


Figure 4-1f

Program Module within Subroutine VALUE to determine λ & μ via:

$\max_{\mu} \min_{\lambda} J(\mu,\lambda)$ ACDYN.92 (Algorithm 2)

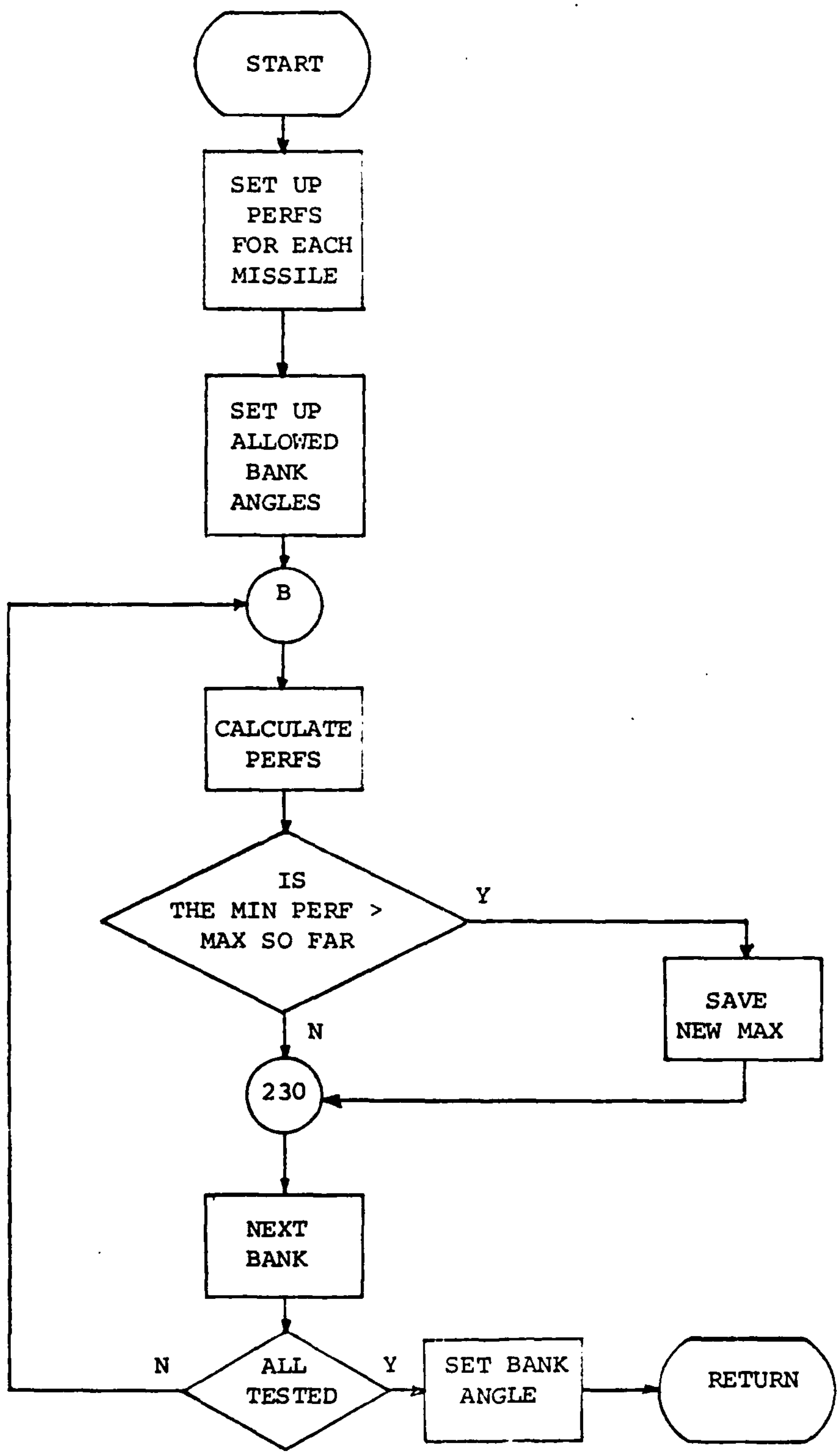


Figure 4-1g

Subroutine <u>INTBOX</u>

(fly one time step)

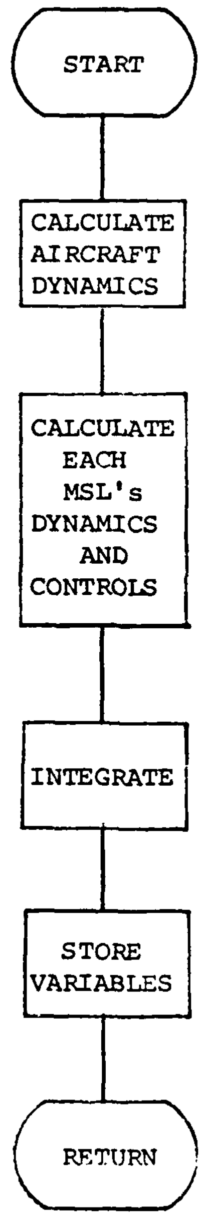
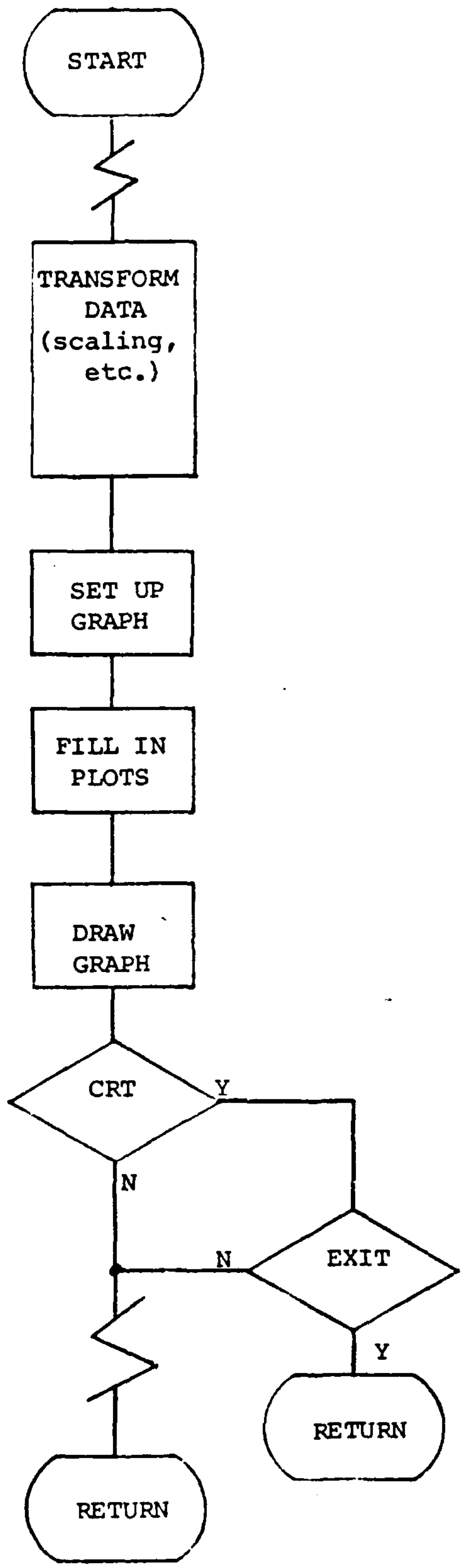


Figure 4-1h

Subroutine PLOUT

(CRT or PRINTER)

(typical use, one set per graph)



Summary of the Initial Conditions of Aircraft & Missiles
at the Beginning of Each Scenario

Scenario #	Vehicle Type:	x ft	y ft	z ft	v ft/sec	γ deg	σ deg
1	A/C	11258	0	30000	1100	0	0
1	MSL 1	11258	14000	33000	3150	-8	-60
1	MSL 2	28578	10000	27000	3450	8	-150
2	A/C	11258	0	30000	1100	0	0
2	MSL 1	11258	15000	33000	3150	-8	-60
2	MSL 2	28578	-10000	27000	3450	8	150
3	A/C	11258	0	30000	1100	0	0
3	MSL 1	0	6500	33000	3150	-8	-30
3	MSL 2	28578	10000	27000	3450	8	-150
4	A/C	11258	0	30000	1100	0	0
4	MSL 1	28258	0	33000	3300	-8	180
4	MSL 2	0	6500	33000	3300	-8	-30
5	A/C	15000	0	30000	1100	0	0
5	MSL 1	3000	0	30000	3300	0	0
5	MSL 2	15000	12000	30000	3300	0	-60
6	A/C	13000	0	30000	1100	0	0
6	MSL 1	30000	0	33000	3300	-8	180
6	MSL 2	0	0	33000	3300	-8	0
7	A/C	0	0	30000	1000	0	45
7	MSL 1	15000	0	30000	3300	0	180
7	MSL 2	0	15000	30000	3300	0	-90

Table 4-1

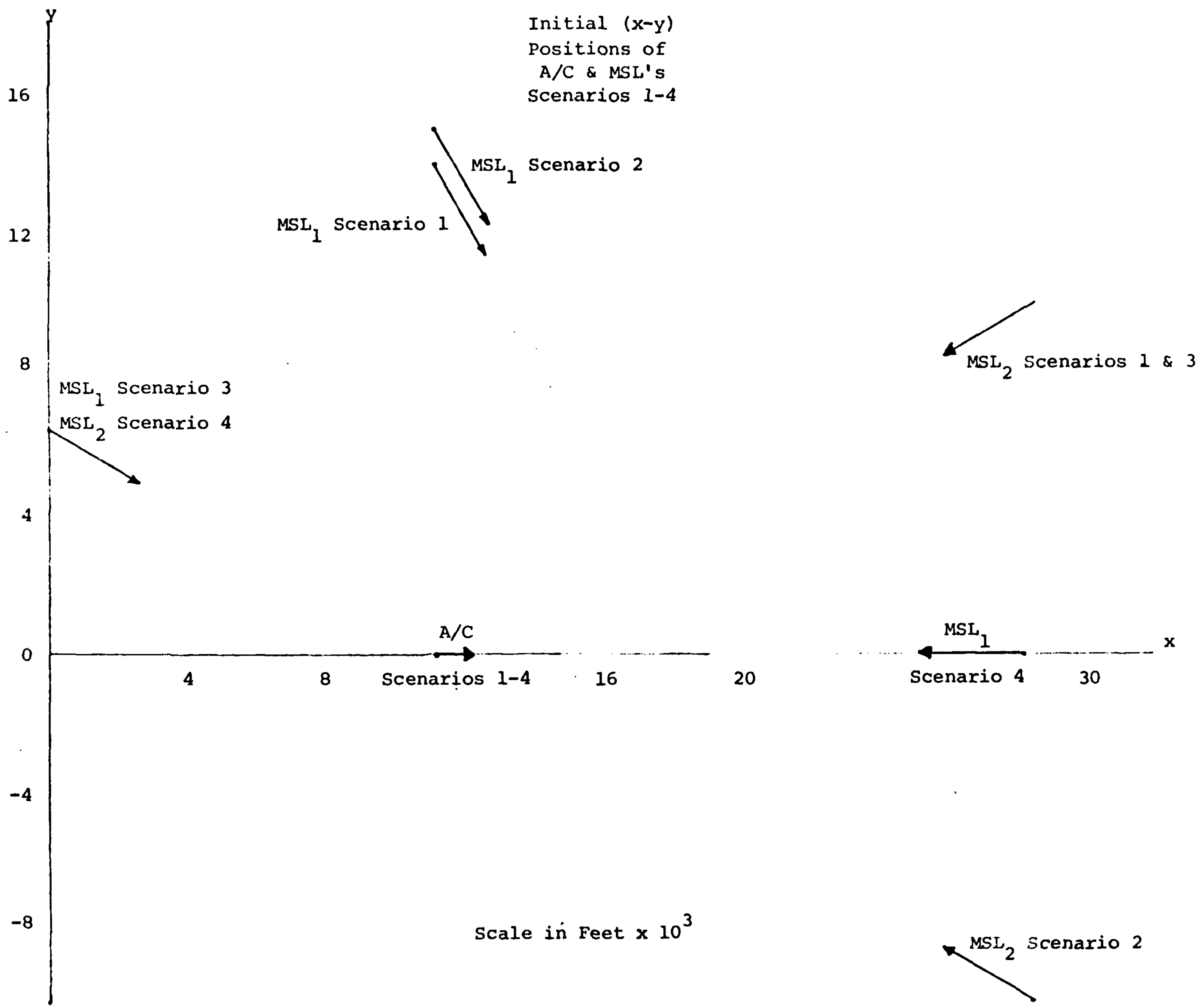

Initial (x-y)
Positions of
A/C & MSL's
Scenarios 1-4
MSL_1 Scenario 2
MSL_1 Scenario 1
MSL_2 Scenarios 1 & 3
MSL_1 Scenario 3
MSL_2 Scenario 4
A/C
Scenarios 1-4
MSL_1
Scenario 4
MSL_2 Scenario 2
Scale in Feet x 10^3
y
x
16
12
8
4
0
-4
-8
4
8
16
20
30

Figure 4-2a

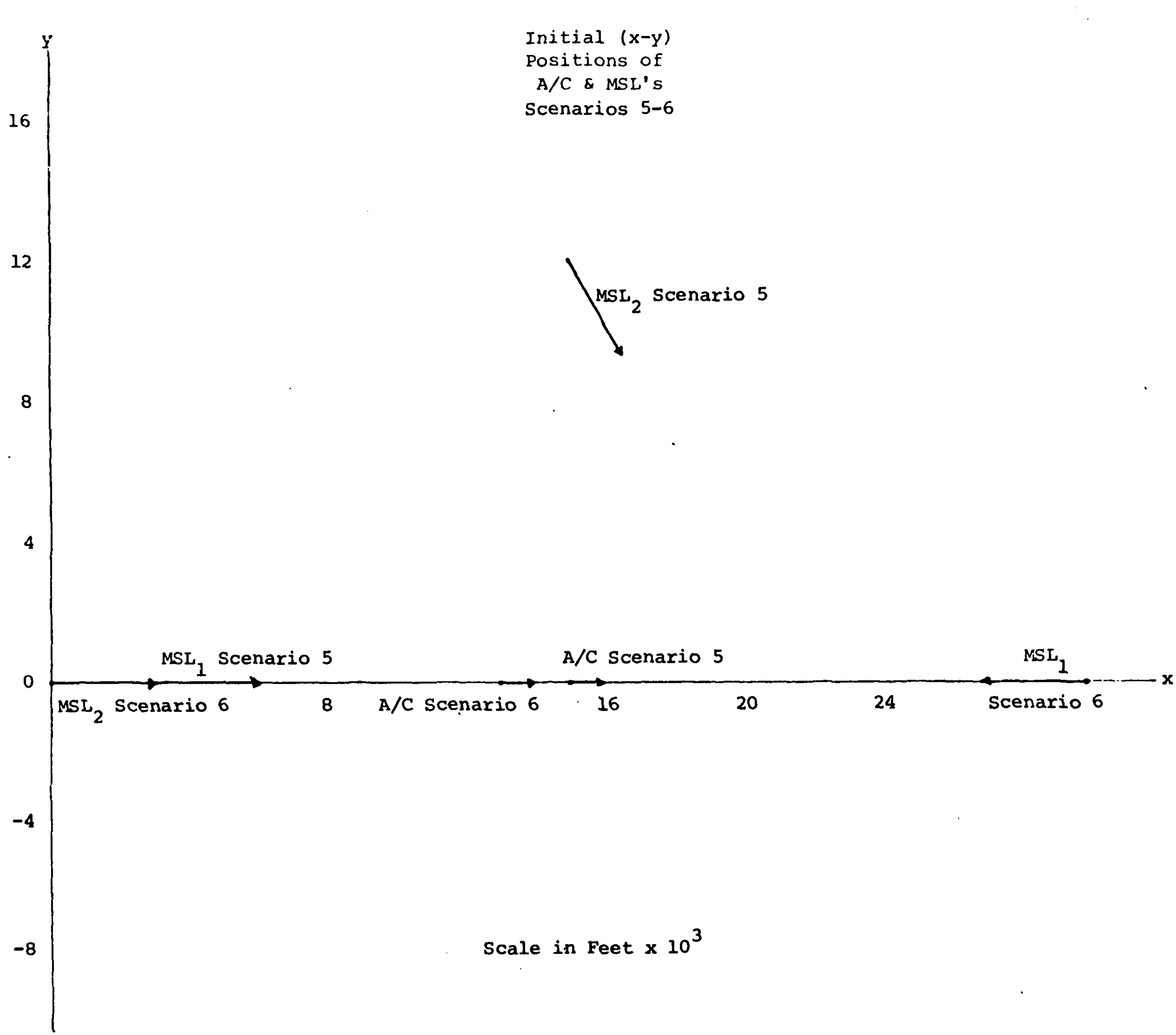

Initial (x-y)
Positions of
A/C & MSL's
Scenarios 5-6
Y
16
12
8
4
0
-4
-8
x
8
16
20
24
MSL1 Scenario 5
MSL2 Scenario 6
A/C Scenario 5
A/C Scenario 6
MSL1
Scenario 6
MSL2 Scenario 5
Scale in Feet x 10^3

Figure 4-2b

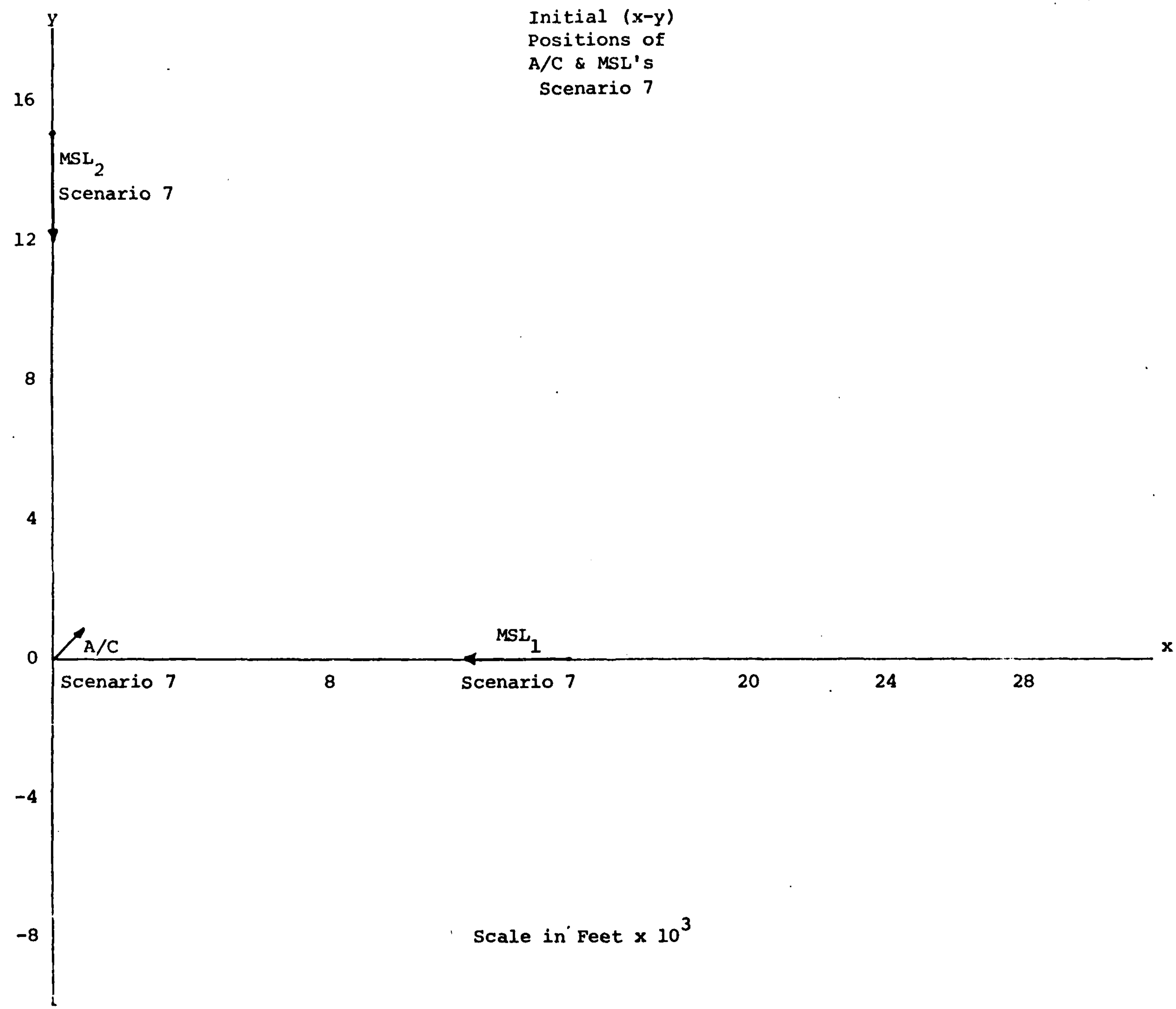

Initial (x-y)
Positions of
A/C & MSL's
Scenario 7
y
16
12
8
4
0
-4
-8
MSL2
Scenario 7
A/C
Scenario 7
MSL1
Scenario 7
8
20
24
28
x
Scale in Feet x 10^3

Figure 4-2c

Table 4-2a

Summary of Algorithm Performance
(Miss Distance)

Scenario #	MSL #	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
1 1	1 2	32 37	18 61	- H	- H
2 2	1 2	40 73	H -	H -	- H
3 3	1 2	T 25	- H	H 30	17 69
4 4	1 2	H -	H -	19 H	21 23
5 5	1 2	29 53	17 52	H 39	- H
6 6	1 2	56 22	60 H	H -	H -
7 7	1 2	64 H	H 25	H -	158 268

Notes: a) Miss distance rounded to nearest foot.

b) H denotes hit, i.e., miss distance $\leq$ 15 feet.

c) - denotes untabulated miss or second hit.

d) T denotes occurrence of time limit in simulation.

Table 4-2b

Definition of Dynamical Variables & Parameters Appearing in Tables 4-3 Through 4-11

State Variables	A/C	MSL_1	MSL_2
x	X0(1)	X0(7)	X0(13)
y	X0(2)	X0(8)	X0(14)
z	X0(3)	X0(9)	X0(15)
v	X0(4)	X0(10)	X0(16)
γ	X0(5)	X0(11)	X0(17)
σ	X0(6)	X0(12)	X0(18)

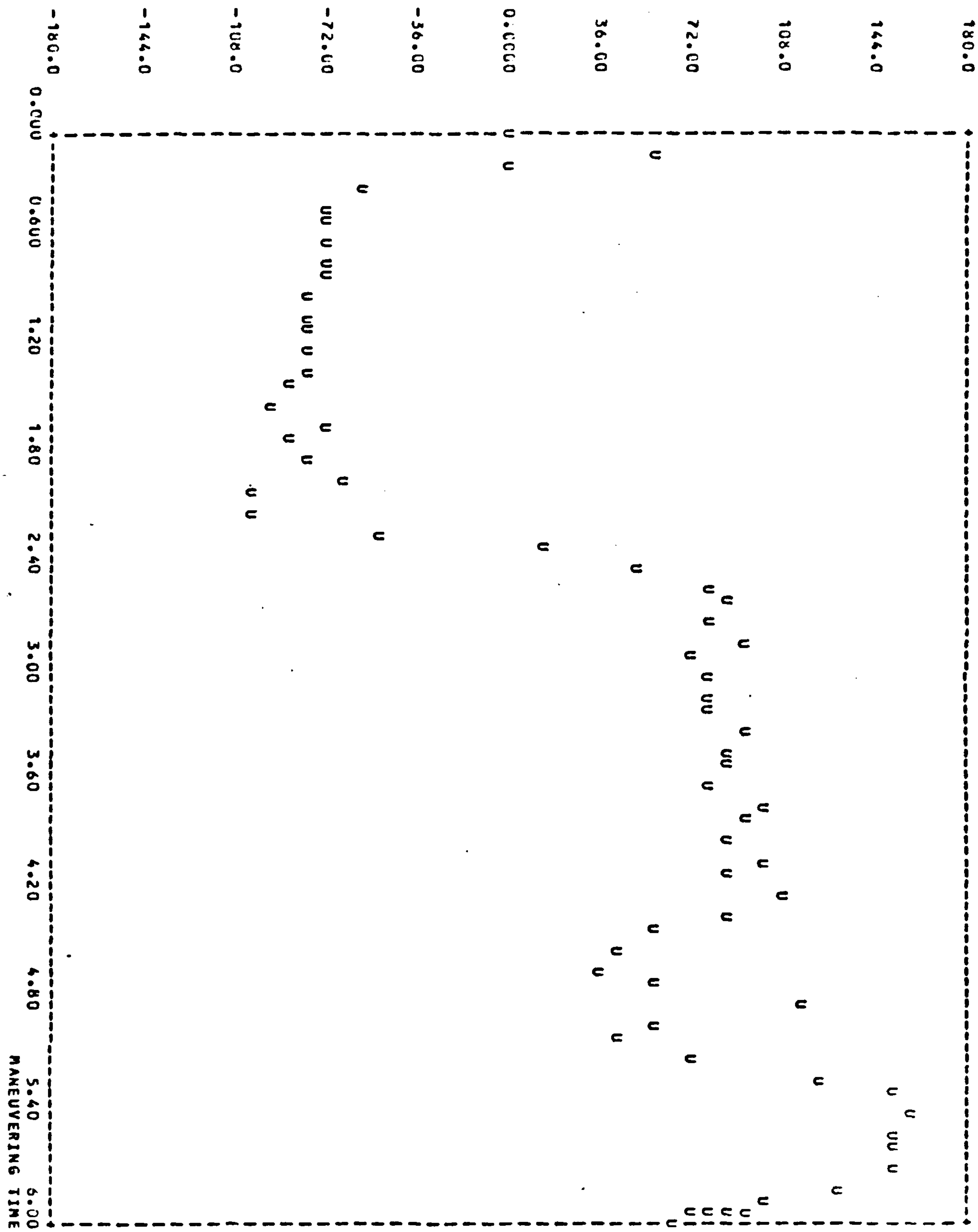
Other Variables/Parameters

Name:	Definition:
DSEP1	Current Distance Between Aircraft and Missile 1
DSEP2	Current Distance Between Aircraft and Missile 2
DMIS	Distance of Closest Approach
TSTEP	Maximum Integration Time Step (see p. 4-2, TSTEP $\triangleq$ T)
TAU	Autopilot-Airframe Time Constant for Missiles 1 & 2
RK11	Proportional Navigation Gain (Pitch) for Missiles 1 & 2
RK12	Proportional Navigation Gain (Yaw) for Missiles 1 & 2

Table 4-3
Algorithm 1, Scenario 1

```
TSTEP =    0.1000, TAU =    0.1500

INIT XO(1) =      11258.0      INIT XO( 7) =      11258.0      INIT XO(13) =      28578.0
INIT XO(2) =      0.000000     INIT XO( 8) =      14000.0      INIT XO(14) =      10000.0
INIT XO(3) =      50000.0      INIT XO( 9) =      33000.0      INIT XO(15) =      27000.0
INIT XO(4) =      1100.00      INIT XO(10) =      3150.00      INIT XO(16) =      3450.00
INIT XO(5) =      0.000000     INIT XO(11) =     -7.99999      INIT XO(17) =      7.99999
INIT XO(6) =      0.000000     INIT XO(12) =     -60.0000      INIT XO(18) =     -150.000
                  PROPORTIONAL NAVIGATION GAINS:
         PITCH (RK11) =     4.50     , YAW (RK12) =     4.50

         A/C MAXIMUM LOAD FACTOR =  6.00

          IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =       0.000  DSEP1 =    0.143E 05  DSEP2 =     0.202E 05
START MANEUVER AT T =    0.10
   TIME =       1.000  DSEP1 =    0.116E 05  DSEP2 =     0.160E 05
   TIME =       2.000  DSEP1 =    0.924E 04  DSEP2 =     0.123E 05
   TIME =       3.000  DSEP1 =    0.706E 04  DSEP2 =     0.908E 04
   TIME =       4.000  DSEP1 =    0.478E 04  DSEP2 =     0.595E 04
   TIME =       5.000  DSEP1 =    0.238E 04  DSEP2 =     0.286E 04
   TIME =       5.924  DSEP1 =     113.      DSEP2 =      116.
  *** CLOSURE RATE NEGATIVE AT TIME =      5.977****
   TA1 : BEST DSEP =     32.3770    , NOW =     38.5233
   TA2 : BEST DSEP =     37.3531    , NOW =     57.9274

XO(1):     0.1672E 05     XO( 7):     0.1673E 05     XO(13):     0.1671E 05
XO(2):    -505.0          XO( 8):    -536.0          XO(14):    -550.5
XO(3):     0.3047E 05     XO( 9):     0.3049E 05     XO(15):     0.3050E 05
XO(4):     752.3          XO(10):     2208.          XO(16):     2175.
XO(5):    -1.643          XO(11):    -14.79          XO(17):     14.34
XO(6):     24.43          XO(12):    -76.61          XO(18):    -146.6

        DELX1:     -8.84        DELY1:      28.0
        DELZ1:     -24.9        DMIS1:      38.5
        BEST DMIS WAS      32.4

        DELX2:      19.3        DELY2:      42.5
        DELZ2:     -34.3        DMIS2:      57.9
        BEST DMIS WAS      37.4
```

Figure 4-3a
Algorithm 1, Scenario 1



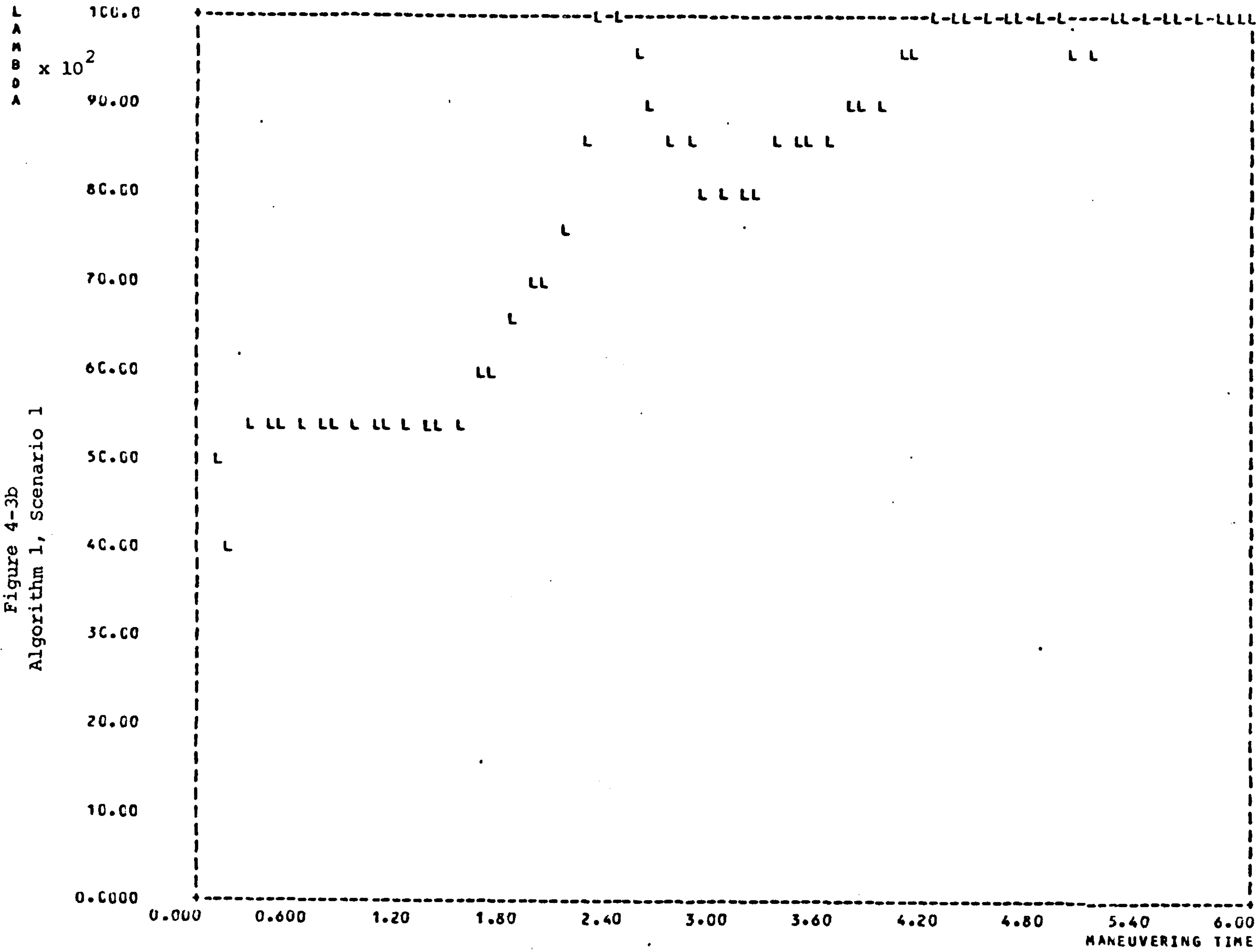


Figure 4-3b
Algorithm 1, Scenario 1

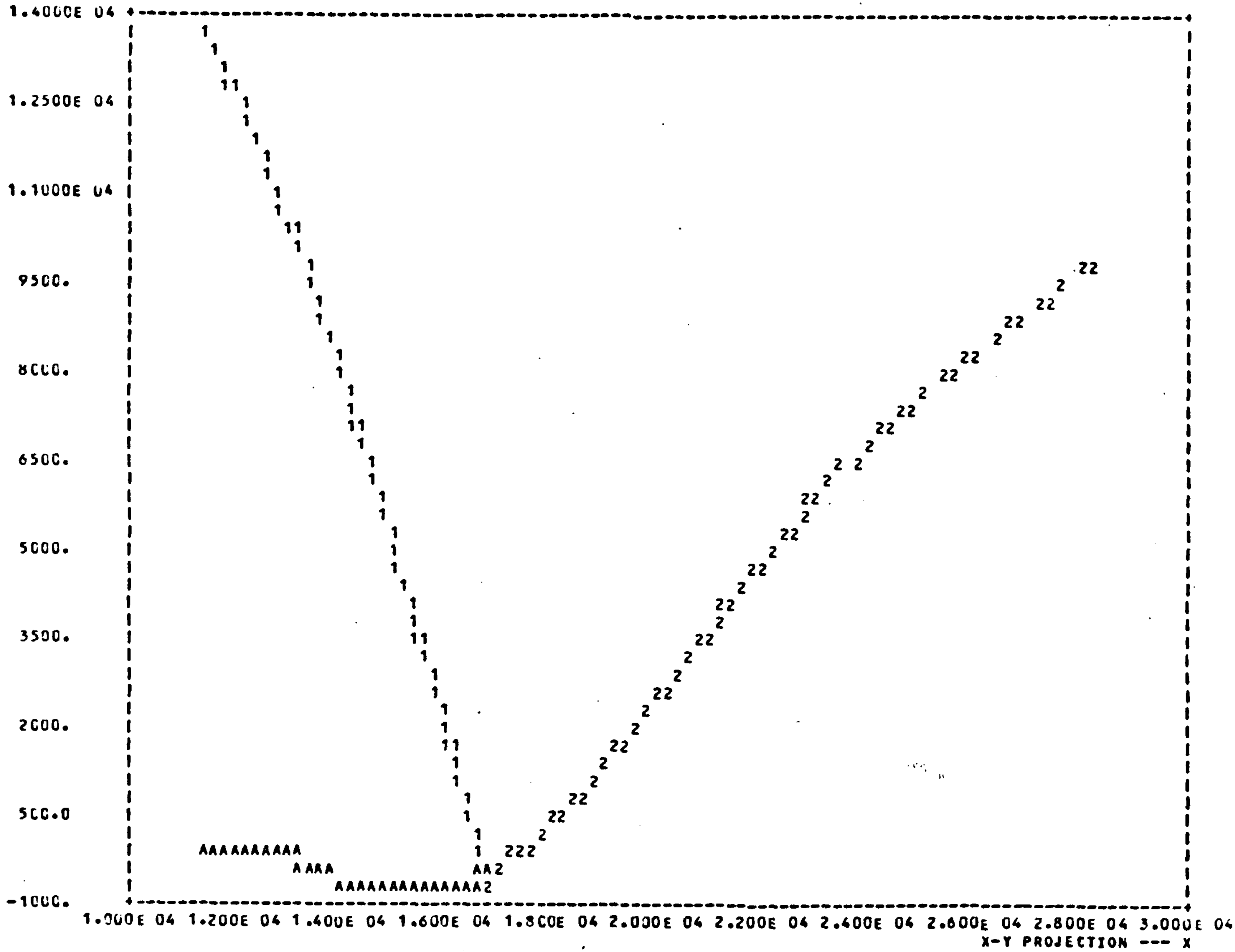


Figure 4-3c
Algorithm 1, Scenario 1

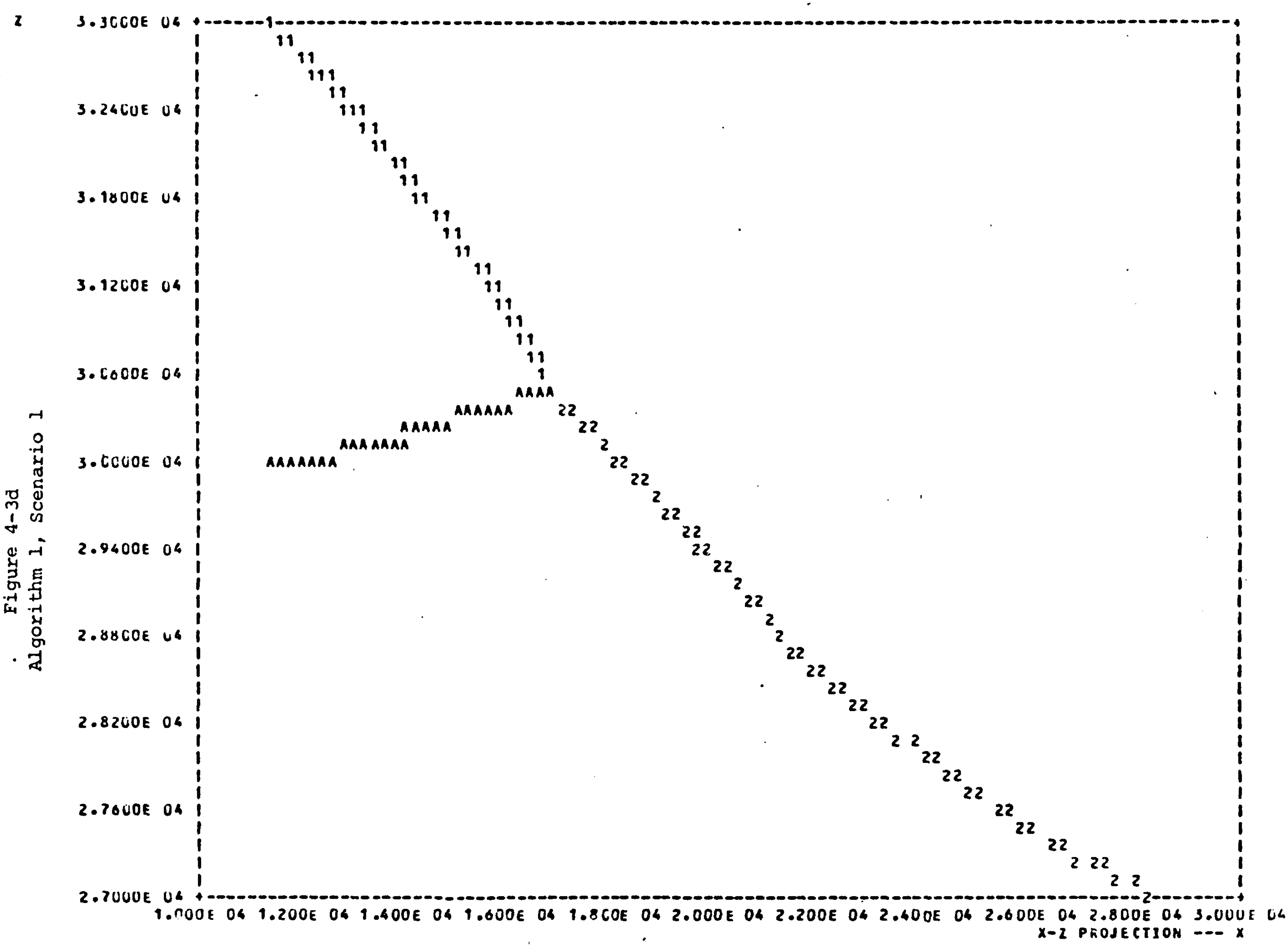


Figure 4-3d
Algorithm 1, Scenario 1

Figure 4-3e
Algorithm 1, Scenario 1

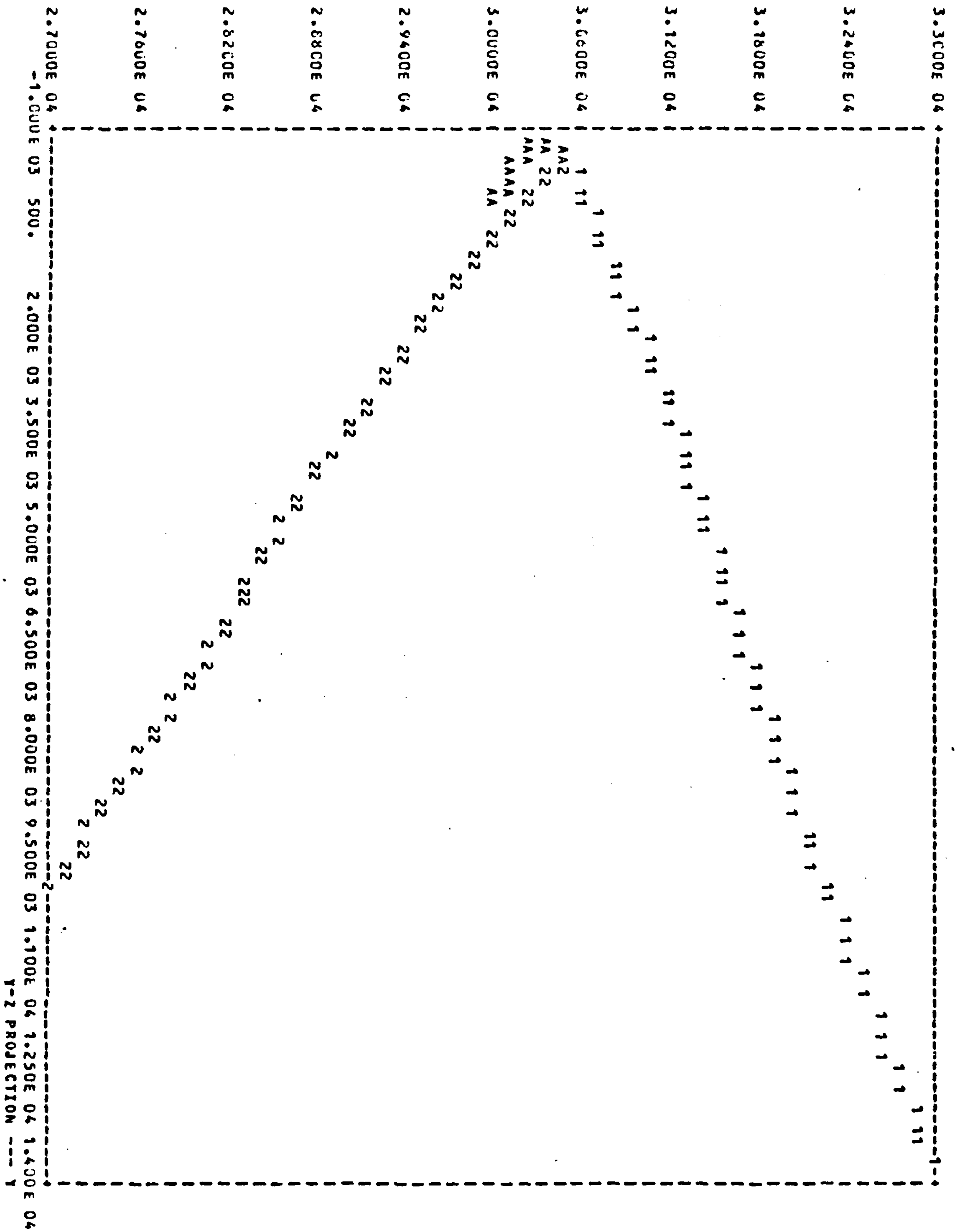
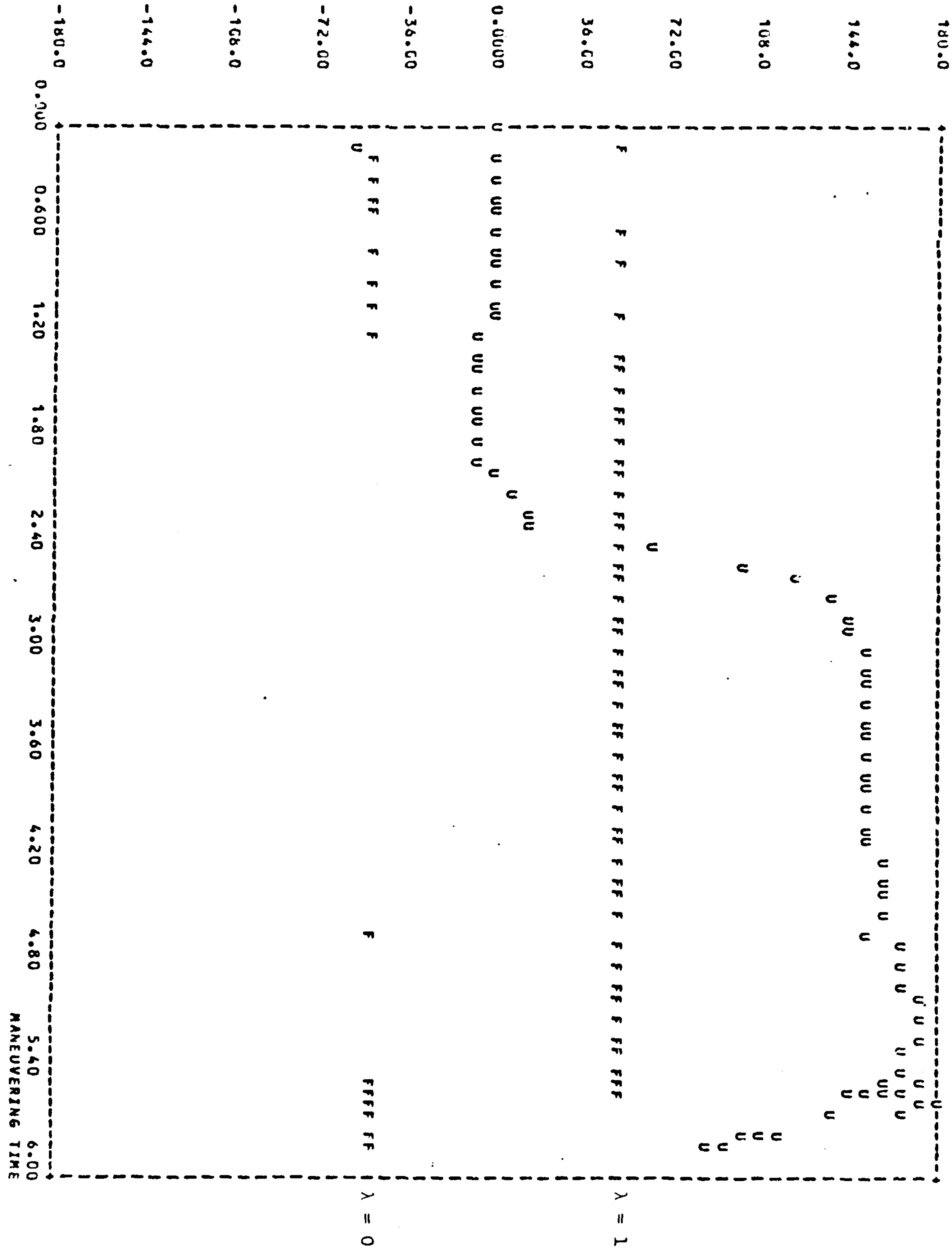


Table 4-4
Algorithm 2, Scenario 1

```
TSTEP =   0.1000, TAU =   0.1500

INIT XO(1) =      11258.0      INIT XO( 7) =      11258.0      INIT XO(13) =      28578.0
INIT XO(2) =      0.000000     INIT XO( 8) =      14000.0      INIT XO(14) =      10000.0
INIT XO(3) =      30000.0      INIT XO( 9) =      33000.0      INIT XO(15) =      27000.0
INIT XO(4) =      1100.00      INIT XO(10) =      3150.00      INIT XO(16) =      3450.00
INIT XO(5) =      0.000000     INIT XO(11) =     -7.99999      INIT XO(17) =      7.99999
INIT XO(6) =      0.000000     INIT XO(12) =     -60.0000      INIT XO(18) =     -150.000
                PROPORTIONAL NAVIGATION GAINS:
         PITCH (RK11) =      4.50     , YAW (RK12) =      4.50

         A/C MAXIMUM LOAD FACTOR =  8.00

          IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =        0.000  DSEP1 =     0.143E 05  DSEP2 =     0.202E 05
START MANEUVER AT T =    0.10
   TIME =        1.000  DSEP1 =     0.116E 05  DSEP2 =     0.160E 05
   TIME =        2.000  DSEP1 =     0.695E 04  DSEP2 =     0.122E 05
   TIME =        3.000  DSEP1 =     0.634E 04  DSEP2 =     0.884E 04
   TIME =        4.000  DSEP1 =     0.373E 04  DSEP2 =     0.557E 04
   TIME =        5.000  DSEP1 =     0.121E 04  DSEP2 =     0.240E 04
   TIME =        5.516  DSEP1 =      21.5       DSEP2 =      830.
   TIME =        5.737  DSEP1 =      629.       DSEP2 =      66.9
 **** CLOSURE RATE NEGATIVE AT TIME =      5.810****
   TA1 : BEST DSEP =      17.8818     , NOW =      600.464
   TA2 : BEST DSEP =      60.9543     , NOW =      69.5214

XO(1):     0.1635E 05     XO( 7):     0.1634E 05     XO(13):     0.1637E 05
XO(2):      625.1         XO( 8):     -30.06         XO(14):      597.1
XO(3):     0.3084E 05     XO( 9):     0.3066E 05     XO(15):     0.3090E 05
XO(4):      750.5         XO(10):      2150.         XO(16):      2182.
XO(5):     -23.99         XO(11):     -24.68         XO(17):      1.543
XO(6):      25.46         XO(12):     -74.65         XO(18):     -147.4

        DELX1:       39.9       DELY1:       655.
        DELZ1:       179.       DMIS1:       680.
        BEST DMIS WAS       17.9

        DELX2:       13.3       DELY2:       28.0
        DELZ2:      -62.2       DMIS2:       69.5
        BEST DMIS WAS       61.0
```

Figure 4-4a
Algorithm 2, Scenario 1



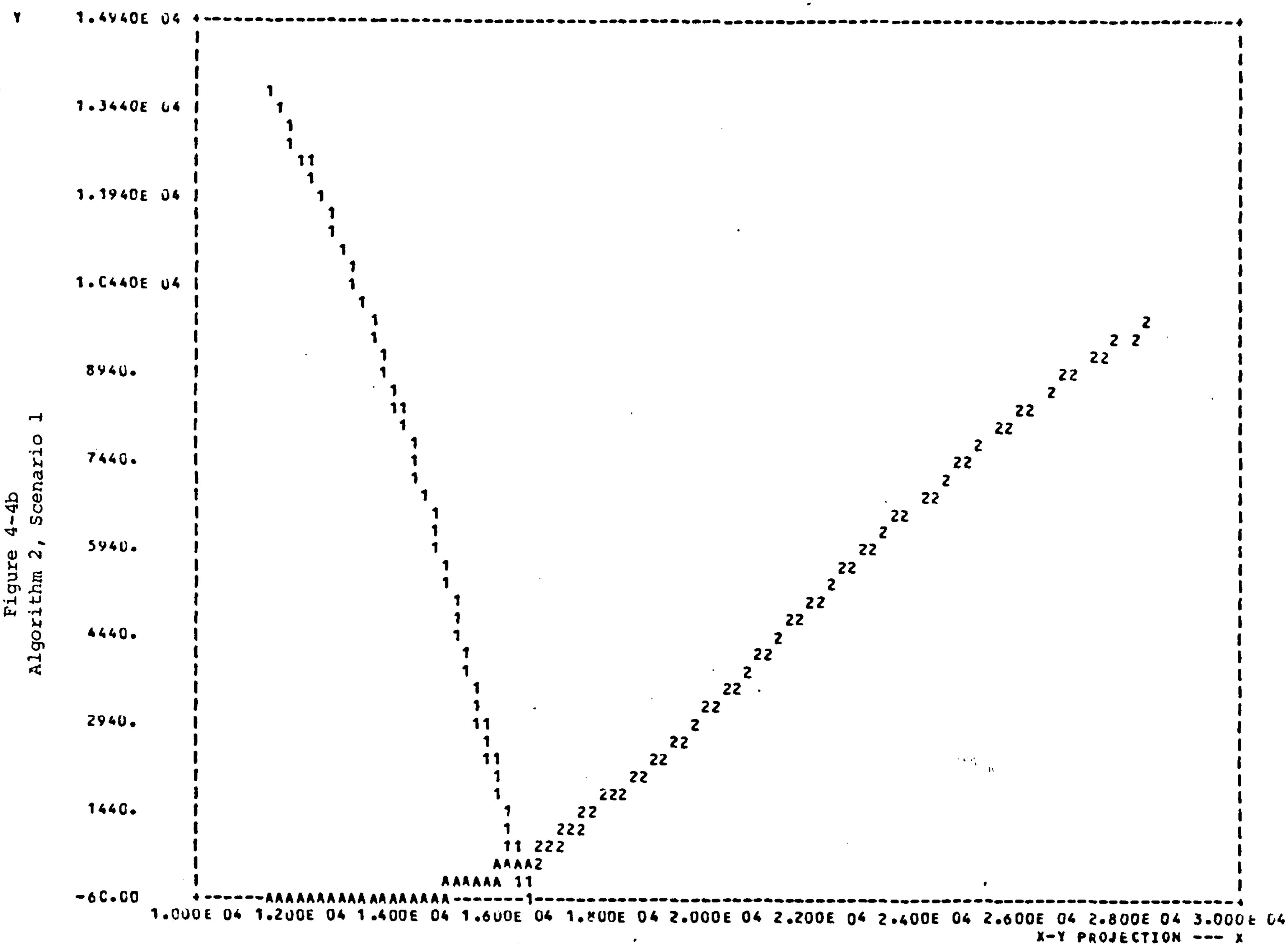


Figure 4-4b
Algorithm 2, Scenario 1

Figure 4-4c
Algorithm 2, Scenario 1

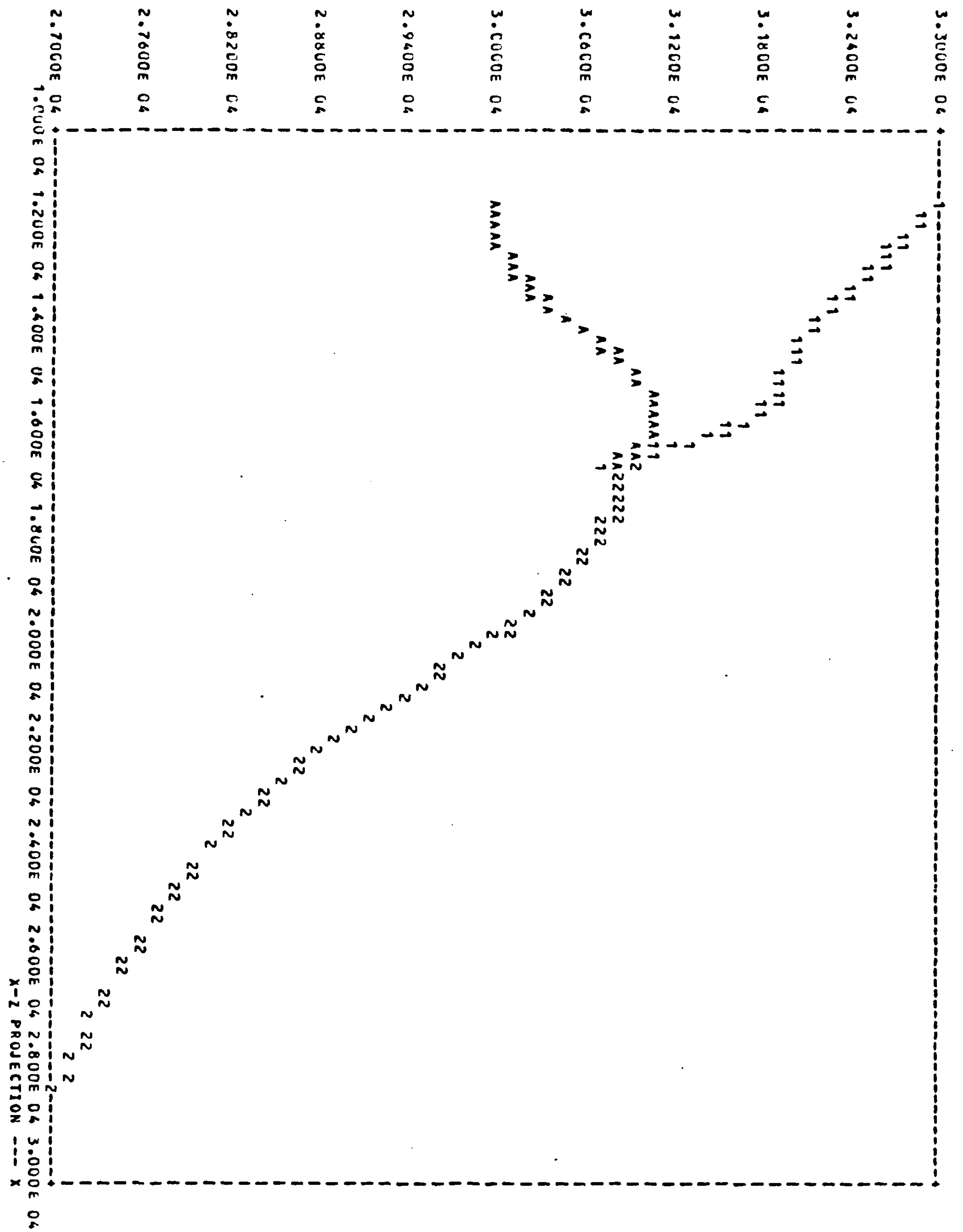


Figure 4-4d
Algorithm 2, Scenario 1

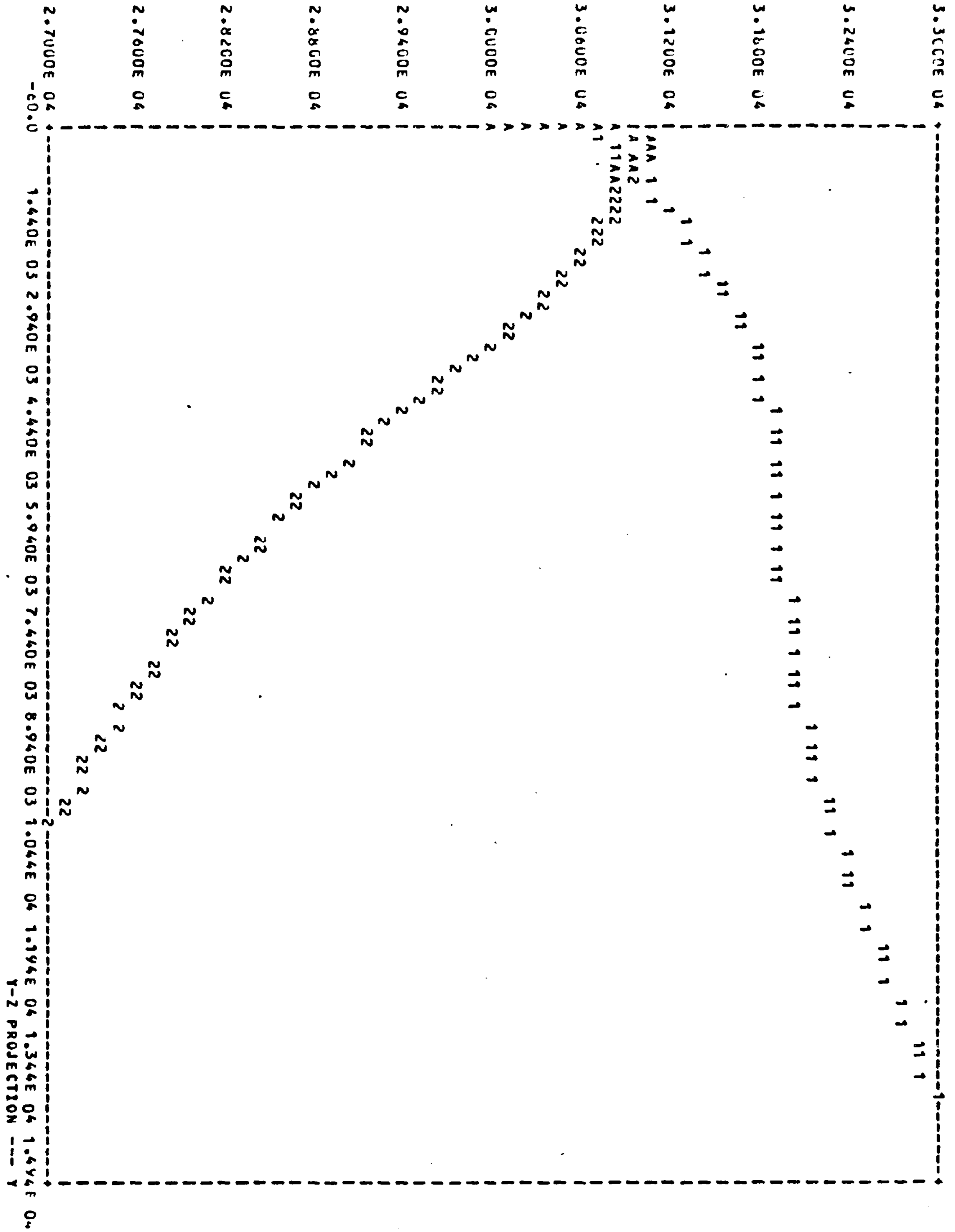


Table 4-5
Algorithm 1, Scenario 2

```
TSTEP =   0.1000, TAU =   0.1500

INIT XO(1) =      11258.0     INIT XO( 7) =      11258.0     INIT XO(13) =     28578.0
INIT XO(2) =      0.000000    INIT XO( 8) =      15000.0     INIT XO(14) =    -9990.00
INIT XO(3) =      [illegible]000.0     INIT XO( 9) =      35000.0     INIT XO(15) =     27000.0
INIT XO(4) =      1100.00     INIT XO(10) =      3150.00     INIT XO(16) =     3450.00
INIT XO(5) =      0.000000    INIT XO(11) =     -7.99999     INIT XO(17) =     7.99999
INIT XO(6) =      0.000000    INIT XO(12) =     -60.0000     INIT XO(18) =     150.000
                 PROPORTIONAL NAVIGATION GAINS:
         PITCH (RK11) =       4.50     , YAW (RK12) =       4.50

         A/C MAXIMUM LOAD FACTOR =  8.00

          IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =        0.000  DSEP1 =    0.153E 05  DSEP2 =    0.202E 05
START MANEUVER AT T =    0.10
   TIME =        1.000  DSEP1 =    0.126E 05  DSEP2 =    0.160E 05
   TIME =        2.000  DSEP1 =    0.980E 04  DSEP2 =    0.122E 05
   TIME =        3.000  DSEP1 =    0.720E 04  DSEP2 =    0.874E 04
   TIME =        4.000  DSEP1 =    0.446E 04  DSEP2 =    0.573E 04
   TIME =        5.000  DSEP1 =    0.170E 04  DSEP2 =    0.310E 04
   TIME =        5.635  DSEP1 =     57.3      DSEP2 =    0.160E 04
   TIME =        [illegible]  DSEP1 =    0.175E 04  DSEP2 =     188.
  *** CLOSURE RATE NEGATIVE AT TIME =        6.376****
   TA1 : BEST DSEP =      40.3800     , NOW =      1973.01
   TA2 : BEST DSEP =      72.7325     , NOW =      73.4026

XO(1):      0.1676E 05     XO( 7):     0.1664E 05     XO(13):     0.1676E 05
XO(2):       2052.         XO( 8):      93.68         XO(14):      2064.
XO(3):      0.2888E 05     XO( 9):     0.2867E 05     XO(15):     0.2896E 05
XO(4):       797.2         XO(10):      2114.         XO(16):      2110.
XO(5):      -19.48         XO(11):     -12.91         XO(17):     0.7152
XO(6):       40.36         XO(12):     -79.90         XO(18):      121.1

         DELX1:      115.       DELY1:     0.196E 04
         DELZ1:      211.       DMIS1:     0.197E 04
         BEST DMIS WAS      40.4

         DELX2:     -1.30       DELY2:     -12.1
         DELZ2:     -72.4       DMIS2:      73.4
         BEST DMIS WAS      72.7
```

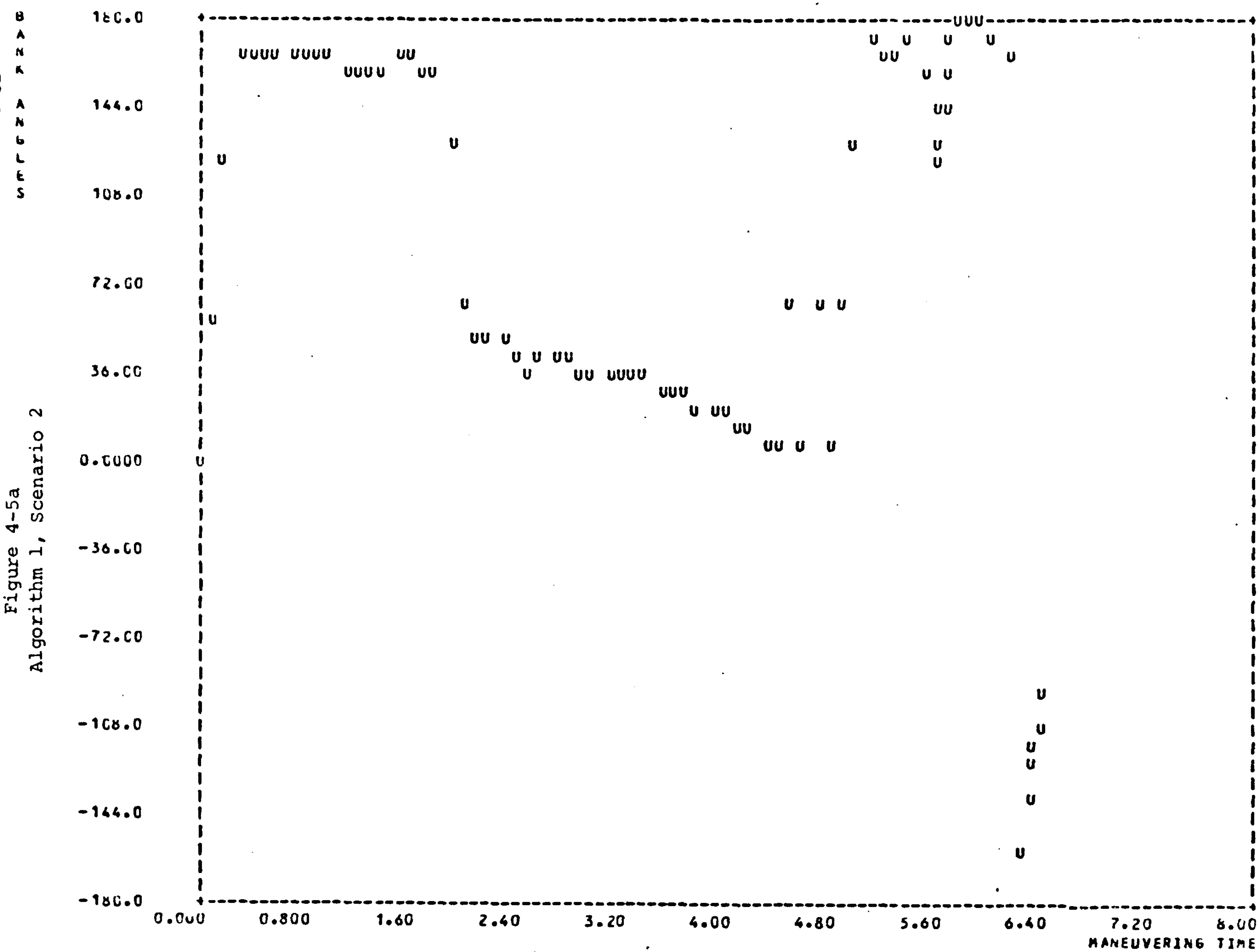


Figure 4-5a
Algorithm 1, Scenario 2

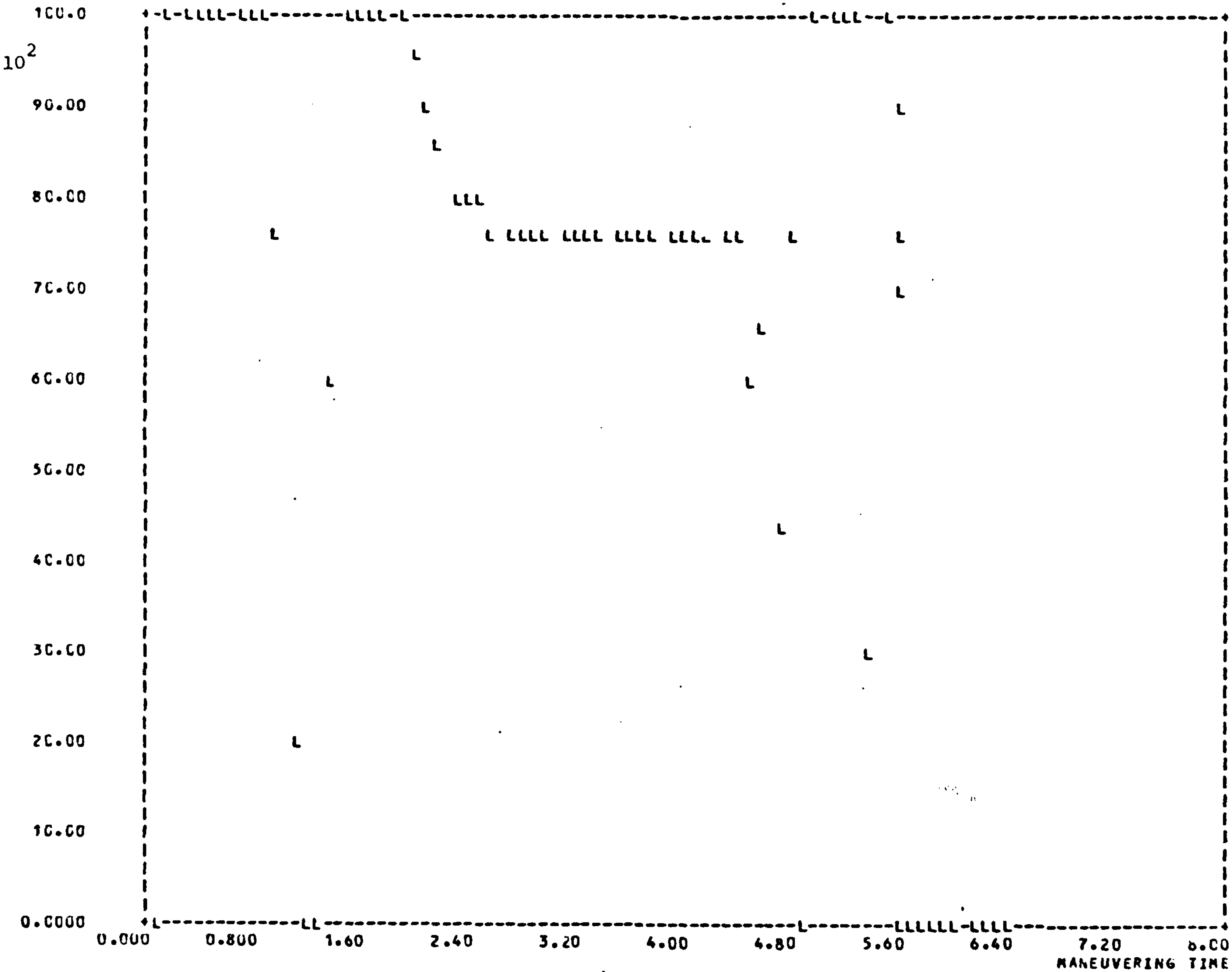


Figure 4-5b
Algorithm 1, Scenario 2

Figure 4-5c
Algorithm 1, Scenario 2

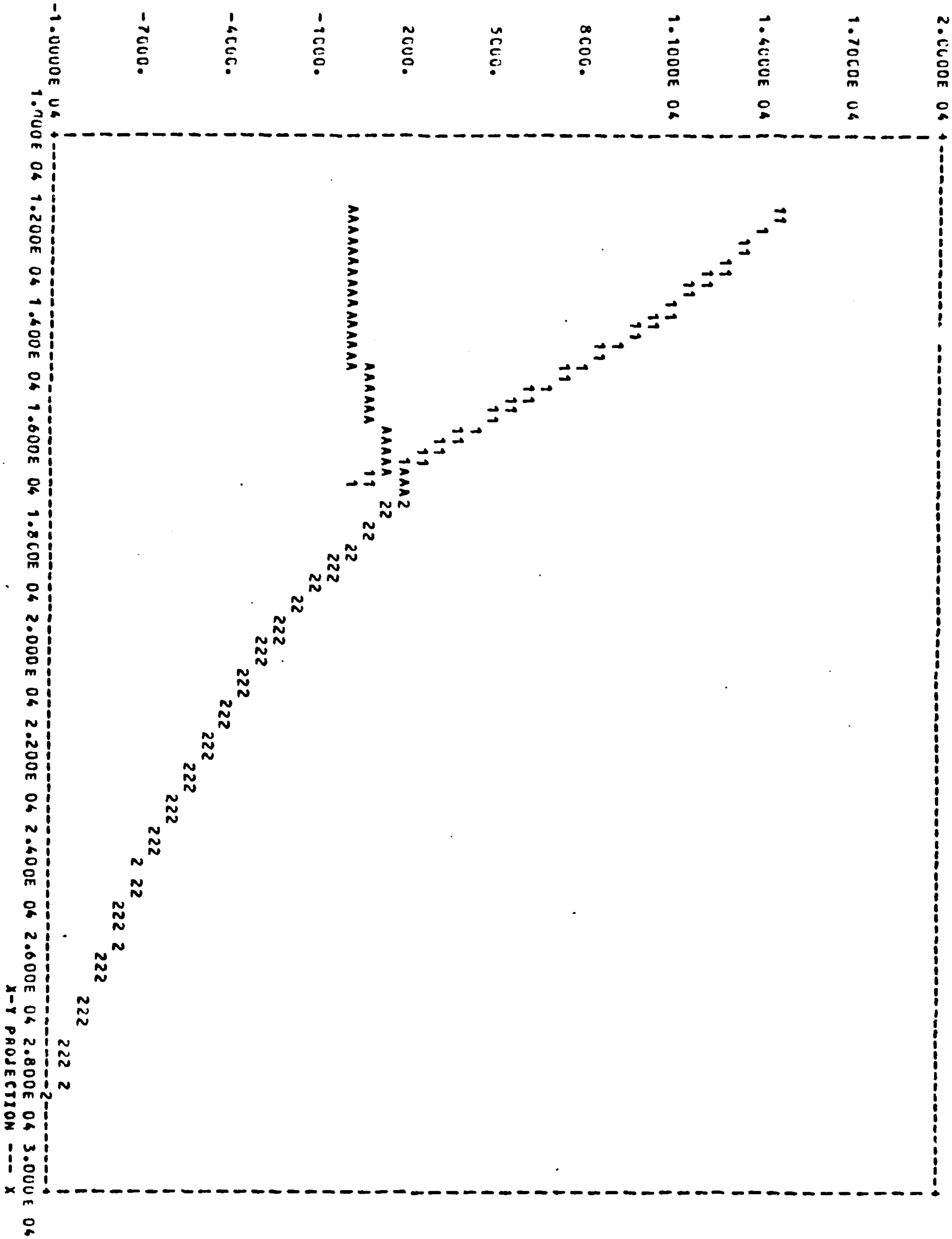
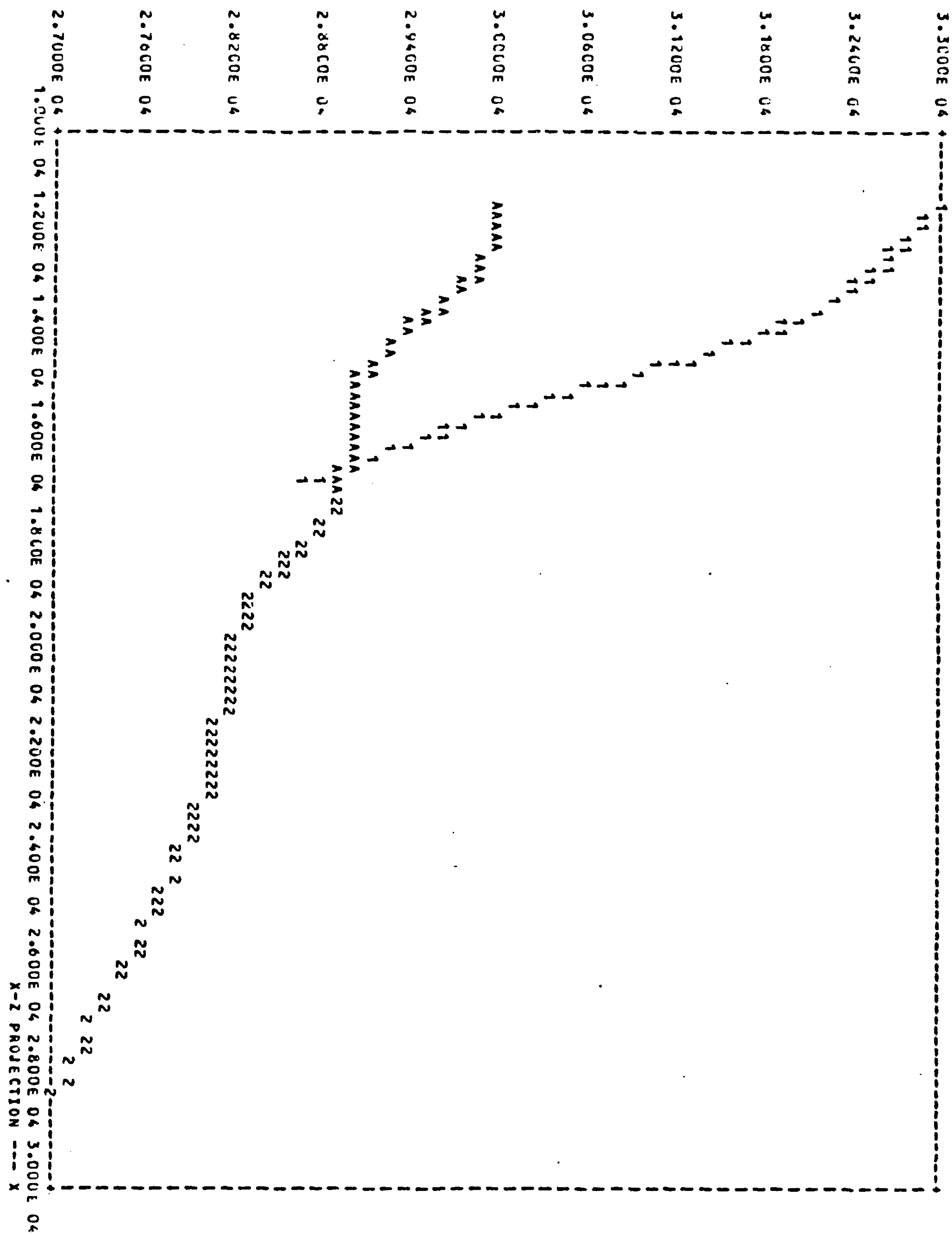


Figure 4-5d
Algorithm 1, Scenario 2



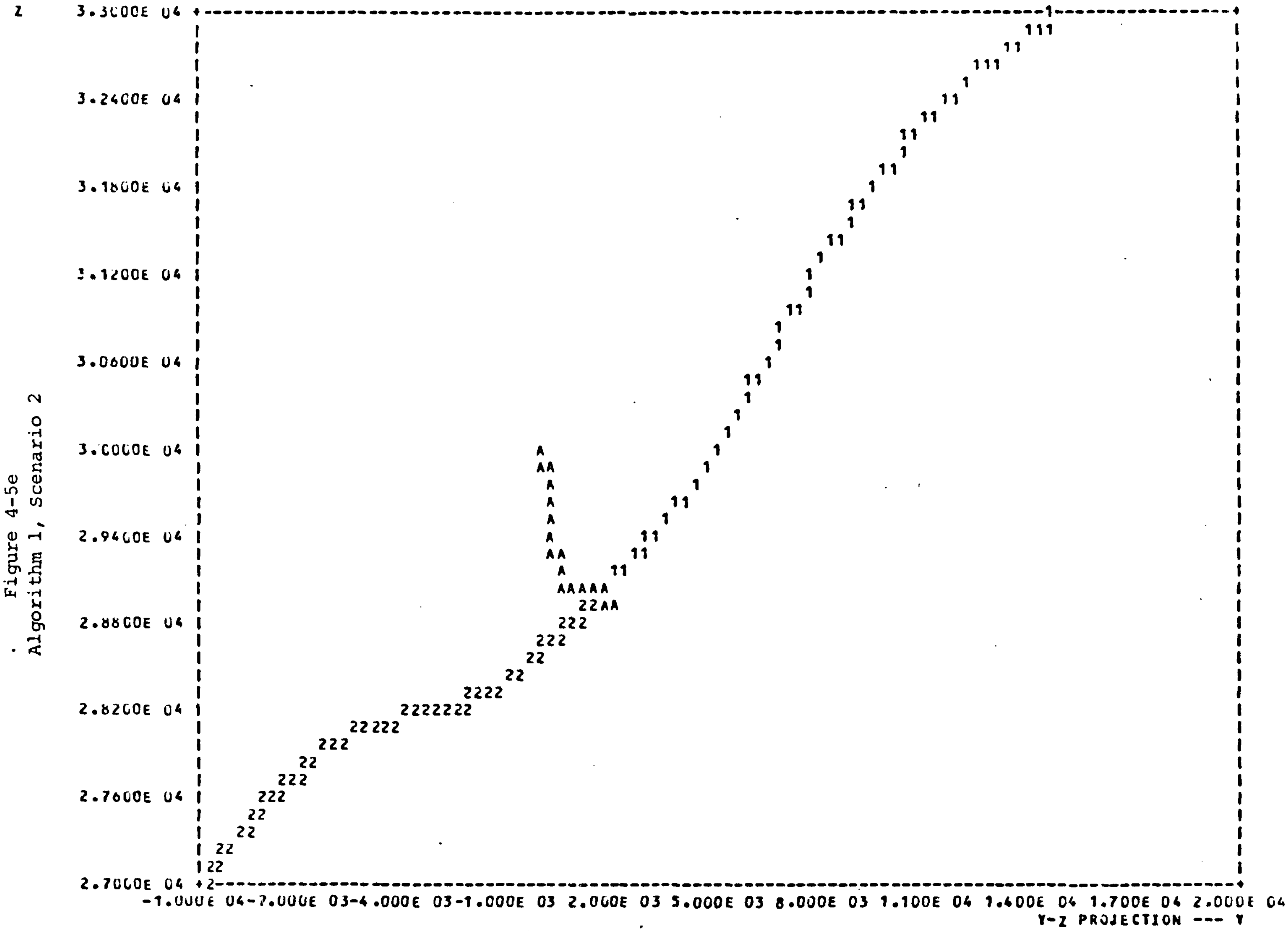


Figure 4-5e
Algorithm 1, Scenario 2

Table 4-6
Algorithm 4, Scenario 3

```
TSTEP =   0.1000, TAU =   0.1500

INIT XO(1) =     11258.0     INIT XO( 7) =    0.000000    INIT XO(13) =     28578.0
INIT XO(2) =    0.000000     INIT XO( 8) =     6500.00    INIT XO(14) =     9990.00
INIT XO(3) =     30000.0     INIT XO( 9) =     33000.0    INIT XO(15) =     27000.0
INIT XO(4) =     1100.00     INIT XO(10) =     3150.00    INIT XO(16) =     3450.00
INIT XO(5) =    0.000000     INIT XO(11) =    -7.99999    INIT XO(17) =     7.99999
INIT XO(6) =    0.000000     INIT XO(12) =    -30.0000    INIT XO(18) =    -150.000
                PROPORTIONAL NAVIGATION GAINS:
        PITCH (RK11) =      4.50    , YAW (RK12) =      4.50

        A/C MAXIMUM LOAD FACTOR =  8.00

         IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =      0.000   DSEP1 =    0.133E 05  DSEP2 =    0.202E 05
START MANEUVER AT T =    0.10
   TIME =      1.000   DSEP1 =    0.112E 05  DSEP2 =    0.159E 05
   TIME =      2.000   DSEP1 =    0.912E 04  DSEP2 =    0.119E 05
   TIME =      3.000   DSEP1 =    0.711E 04  DSEP2 =    0.607E 04
   TIME =      4.000   DSEP1 =    0.515E 04  DSEP2 =    0.457E 04
   TIME =      5.000   DSEP1 =    0.326E 04  DSEP2 =    0.150E 04
   TIME =      5.566   DSEP1 =    0.226E 04  DSEP2 =     76.9
   TIME =      6.126   DSEP1 =    0.135E 04  DSEP2 =    0.144E 04
   TIME =      7.006   DSEP1 =     49.6      DSEP2 =    0.356E 04
 **** CLOSURE RATE NEGATIVE AT TIME =       7.045****
   TA1 : BEST DSEP =      17.3176     , NOW =      18.4981
   TA2 : BEST DSEP =      68.6587     , NOW =      3651.30

XO(1):    0.1586E 05     XO( 7):    0.1587E 05    XO(13):    0.1249E 05
XO(2):     2601.         XO( 8):     2612.        XO(14):     1272.
XO(3):    0.2728E 05     XO( 9):    0.2727E 05    XO(15):    0.2774E 05
XO(4):     797.2         XO(10):     2016.        XO(16):     2035.
XO(5):    -62.09         XO(11):    -35.11        XO(17):    -12.39
XO(6):    -4.609         XO(12):    -23.37        XO(18):    -161.9

        DELX1:     -9.12       DELY1:     -10.8
        DELZ1:      11.9       DMIS1:      18.5
        BEST DMIS WAS      17.3

        DELX2:     0.337E 04   DELY2:     0.133E 04
        DELZ2:     -458.       DMIS2:     0.365E 04
        BEST DMIS WAS      68.7
```

Figure 4-6a
Algorithm 4, Scenario 3

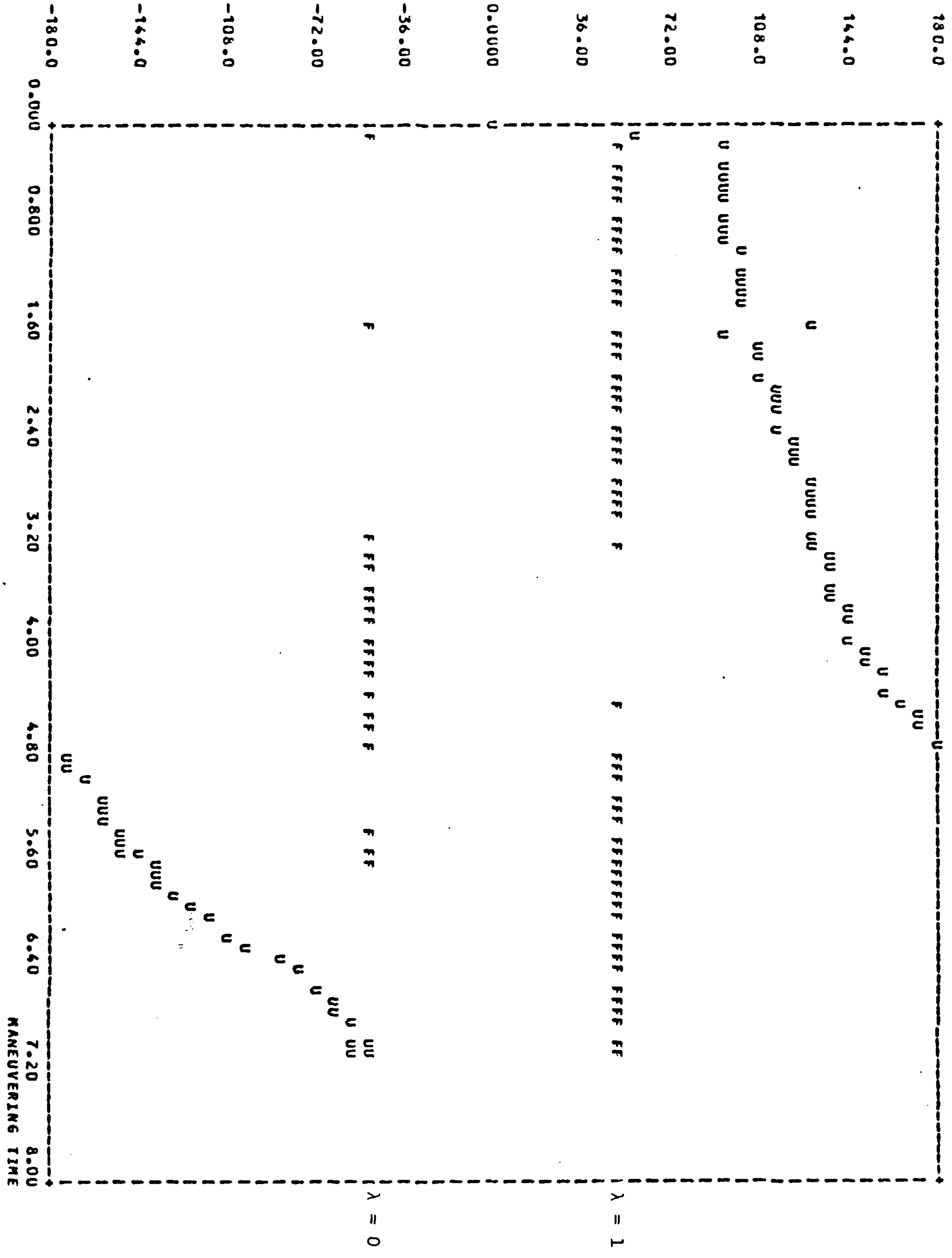
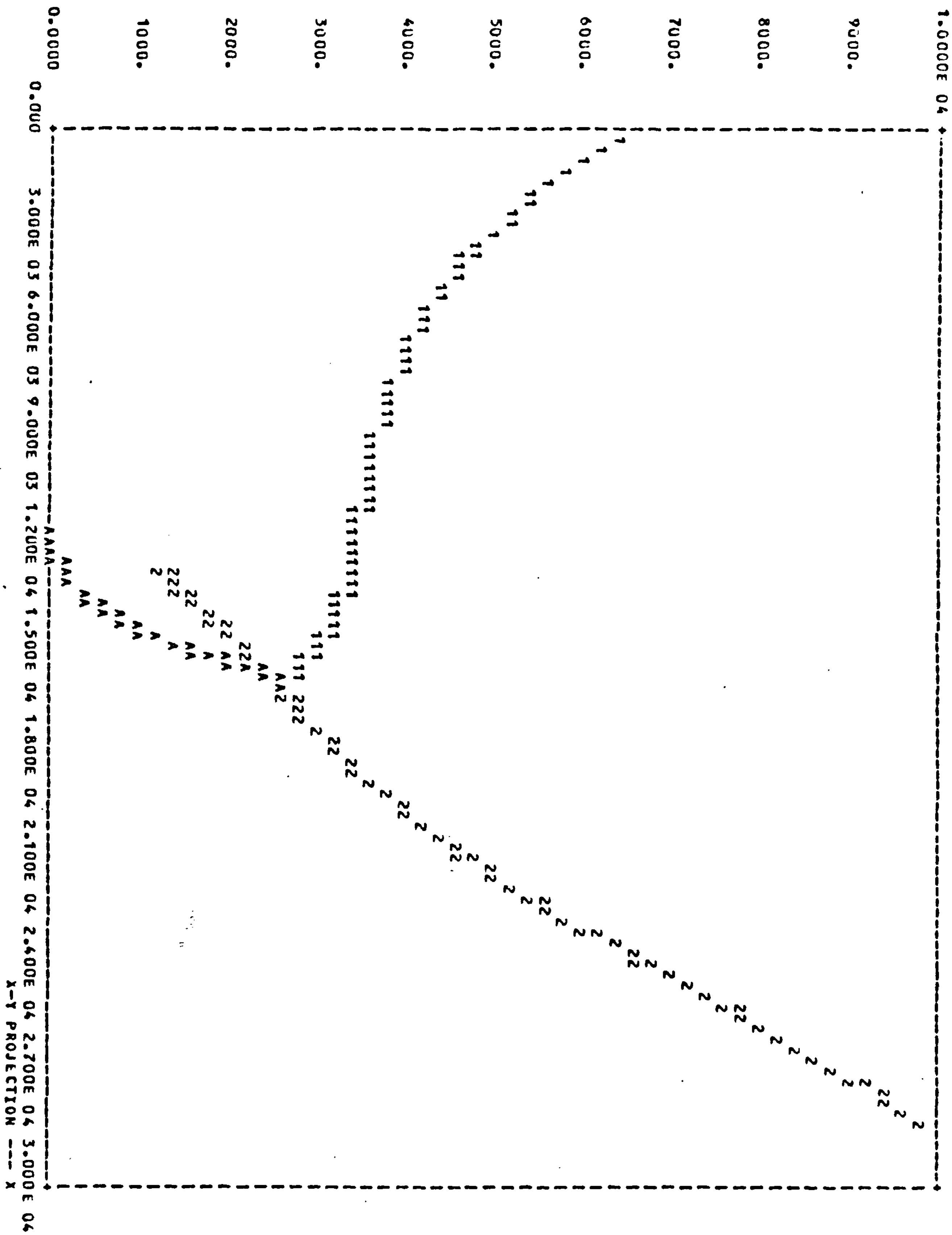


Figure 4-6b
Algorithm 4, Scenario 3



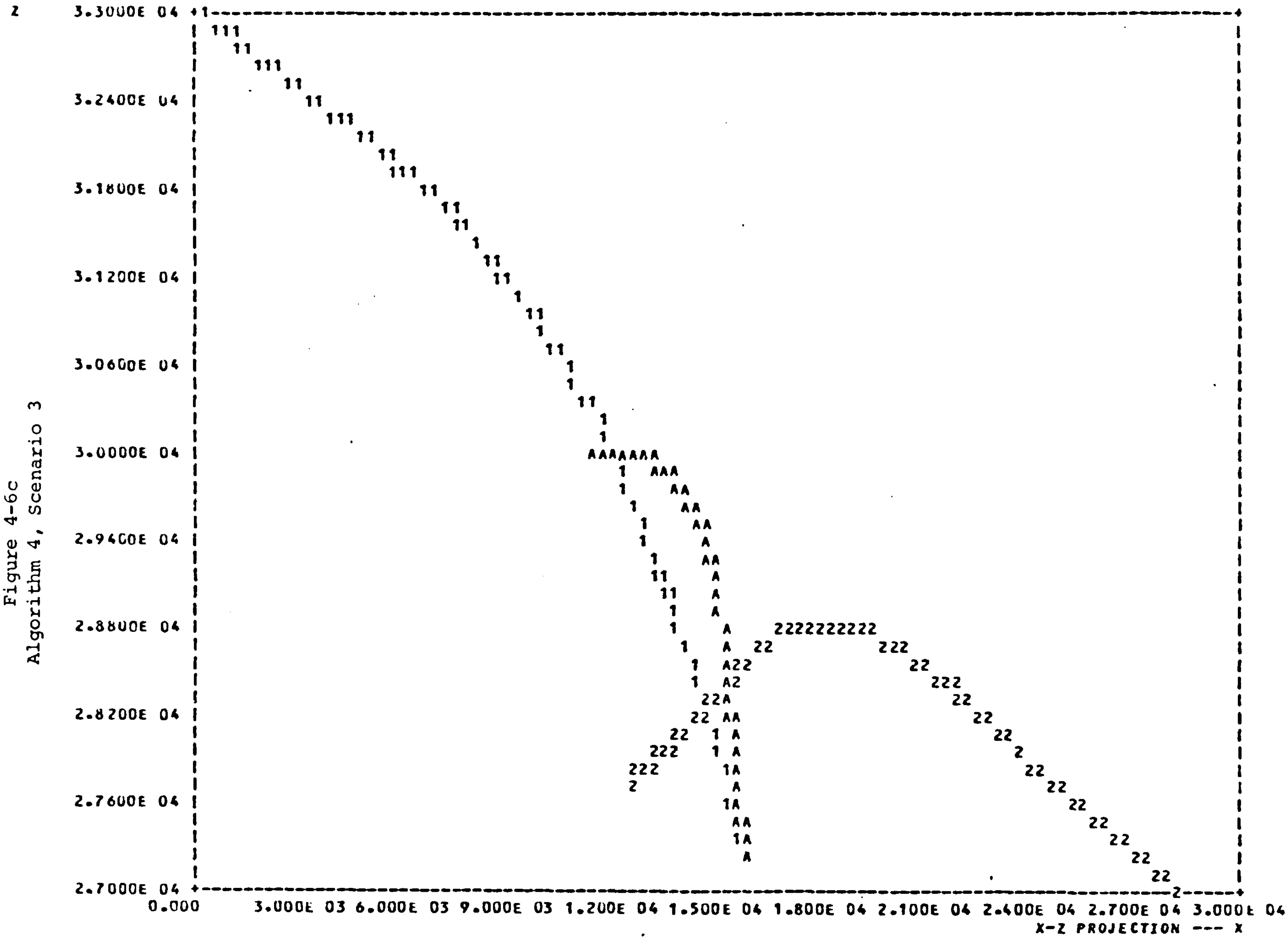


Figure 4-6c
Algorithm 4, Scenario 3

Figure 4-6d
Algorithm 4, Scenario 3

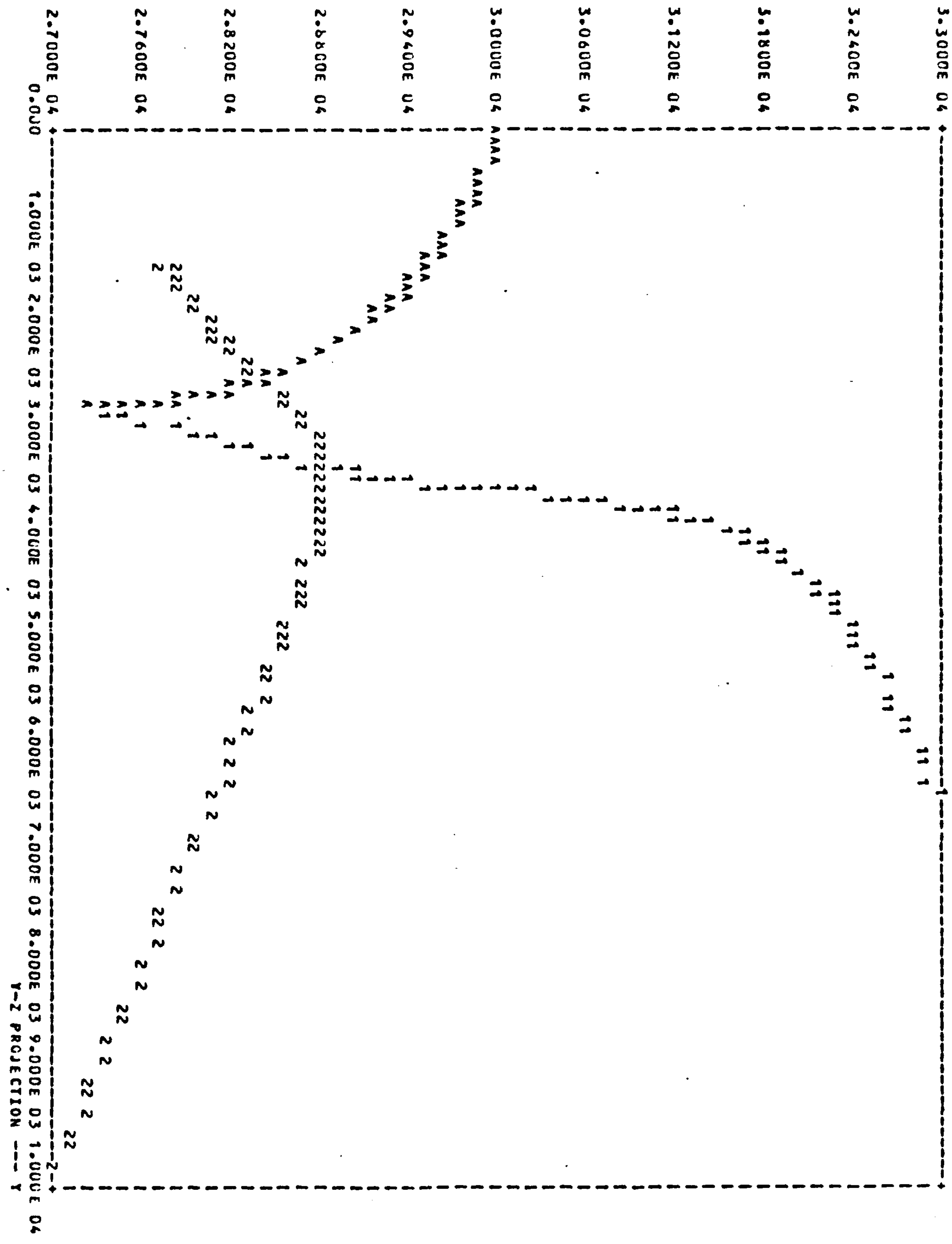


Table 4-7
Algorithm 4, Scenario 4

```
ISTEP =    0.1000, TAU =    0.1500

INIT XO(1) =     11258.0     INIT XO( 7) =     28258.0     INIT XO(13) =     0.000000
INIT XO(2) =     0.000000    INIT XO( 8) =     0.000000    INIT XO(14) =     6500.00
INIT XO(3) =     30000.0     INIT XO( 9) =     33000.0     INIT XO(15) =     33000.0
INIT XO(4) =     1100.00     INIT XO(10) =     3300.00     INIT XO(16) =     3300.00
INIT XO(5) =     0.000000    INIT XO(11) =     -7.99999    INIT XO(17) =     -7.99999
INIT XO(6) =     0.000000    INIT XO(12) =     160.000     INIT XO(18) =     -30.0000
                  PROPORTIONAL NAVIGATION GAINS:
       PITCH (RK11) =      4.50    , YAW (RK12) =      4.50

       A/C MAXIMUM LOAD FACTOR =  8.00

        IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =        0.000   DSEP1 =     0.173E 05   DSEP2 =     0.133E 05
START MANEUVER AT T =    0.10
   TIME =        1.000   DSEP1 =     0.130E 05   DSEP2 =     0.111E 05
   TIME =        2.000   DSEP1 =     0.910E 04   DSEP2 =     0.889E 04
   TIME =        3.000   DSEP1 =     0.563E 04   DSEP2 =     0.681E 04
   TIME =        4.000   DSEP1 =     0.269E 04   DSEP2 =     0.479E 04
   TIME =        4.970   DSEP1 =     274.        DSEP2 =     0.293E 04
   TIME =        5.096   DSEP1 =     29.8        DSEP2 =     0.270E 04
   TIME =        5.360   DSEP1 =     633.        DSEP2 =     0.225E 04
   TIME =        6.360   DSEP1 =     0.287E 04   DSEP2 =     636.
 **** CLOSURE RATE NEGATIVE AT TIME =      6.795****
   TA1 : BEST DSEP =      20.5991     , NOW =      3791.88
   TA2 : BEST DSEP =      23.0974     , NOW =      25.1589

XO(1):    0.1586E 05     XO( 7):    0.1225E 05     XO(13):    0.1590E 05
XO(2):     2139.         XO( 8):     2707.         XO(14):     2156.
XO(3):    0.2693E 05     XO( 9):    0.2598E 05     XO(15):    0.2692E 05
XO(4):     846.0         XO(10):     2124.         XO(16):     2126.
XO(5):    -64.77         XO(11):    -37.31         XO(17):    -36.24
XO(6):    -19.29         XO(12):     158.7         XO(18):    -24.84

       DELX1:    0.363E 04   DELY1:     -568.
       DELZ1:     949.       DMIS1:     0.379E 04
       BEST DMIS WAS      20.6

       DELX2:    -17.2       DELY2:     -16.8
       DELZ2:     7.29       DMIS2:      25.2
       BEST DMIS WAS      23.1
```

Figure 4-7a

Algorithm 4, Scenario 4

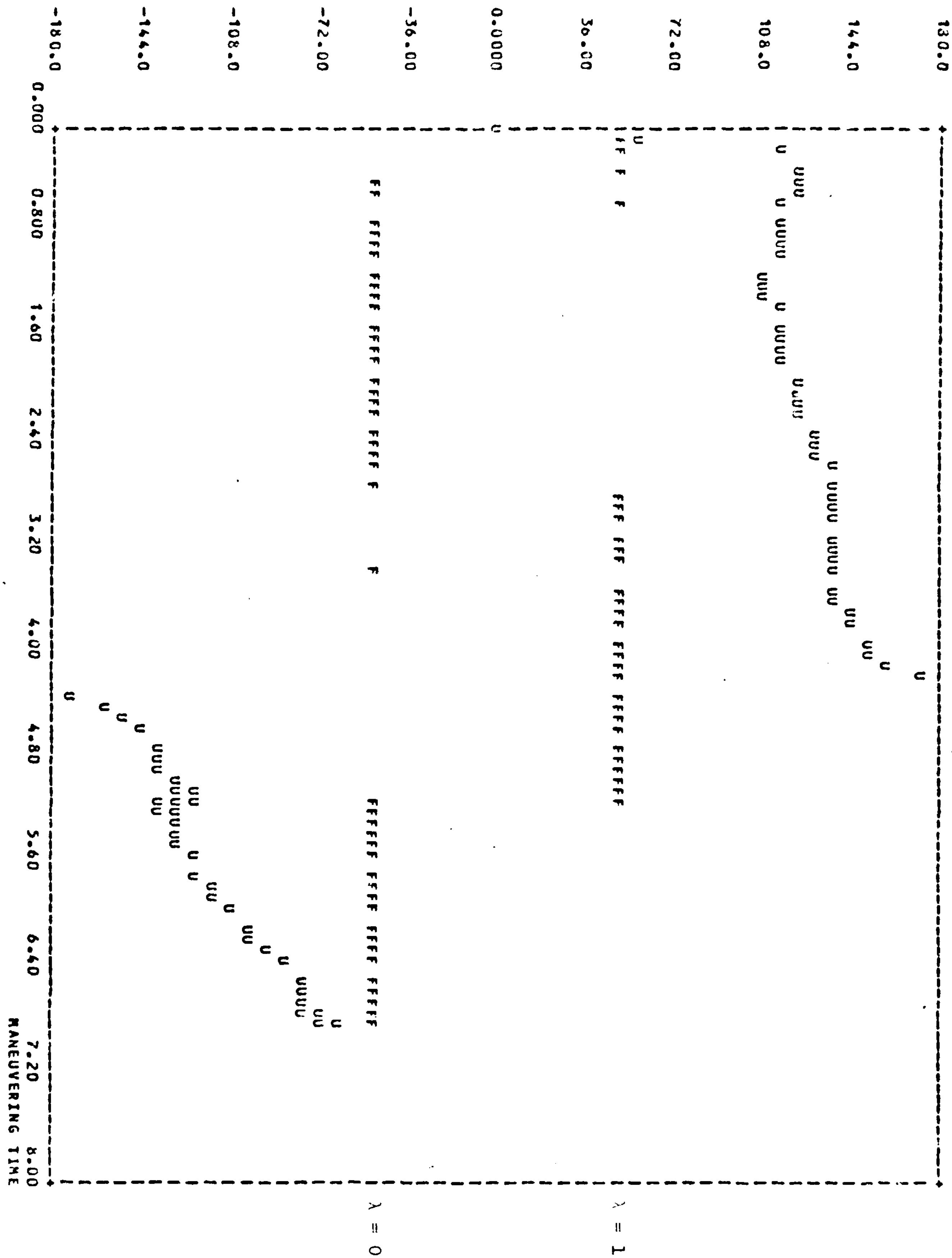


Figure 4-7b
Algorithm 4, Scenario 4

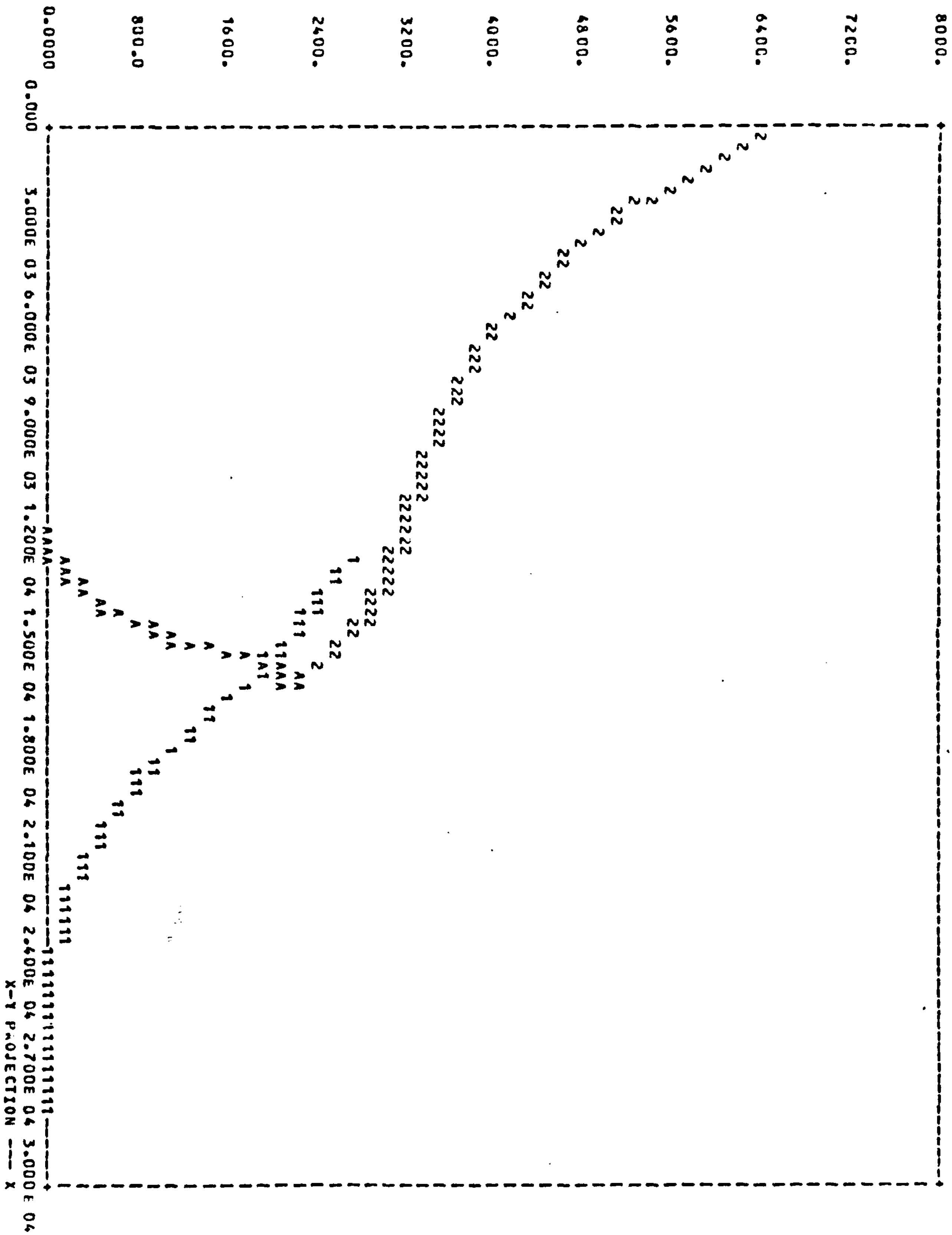
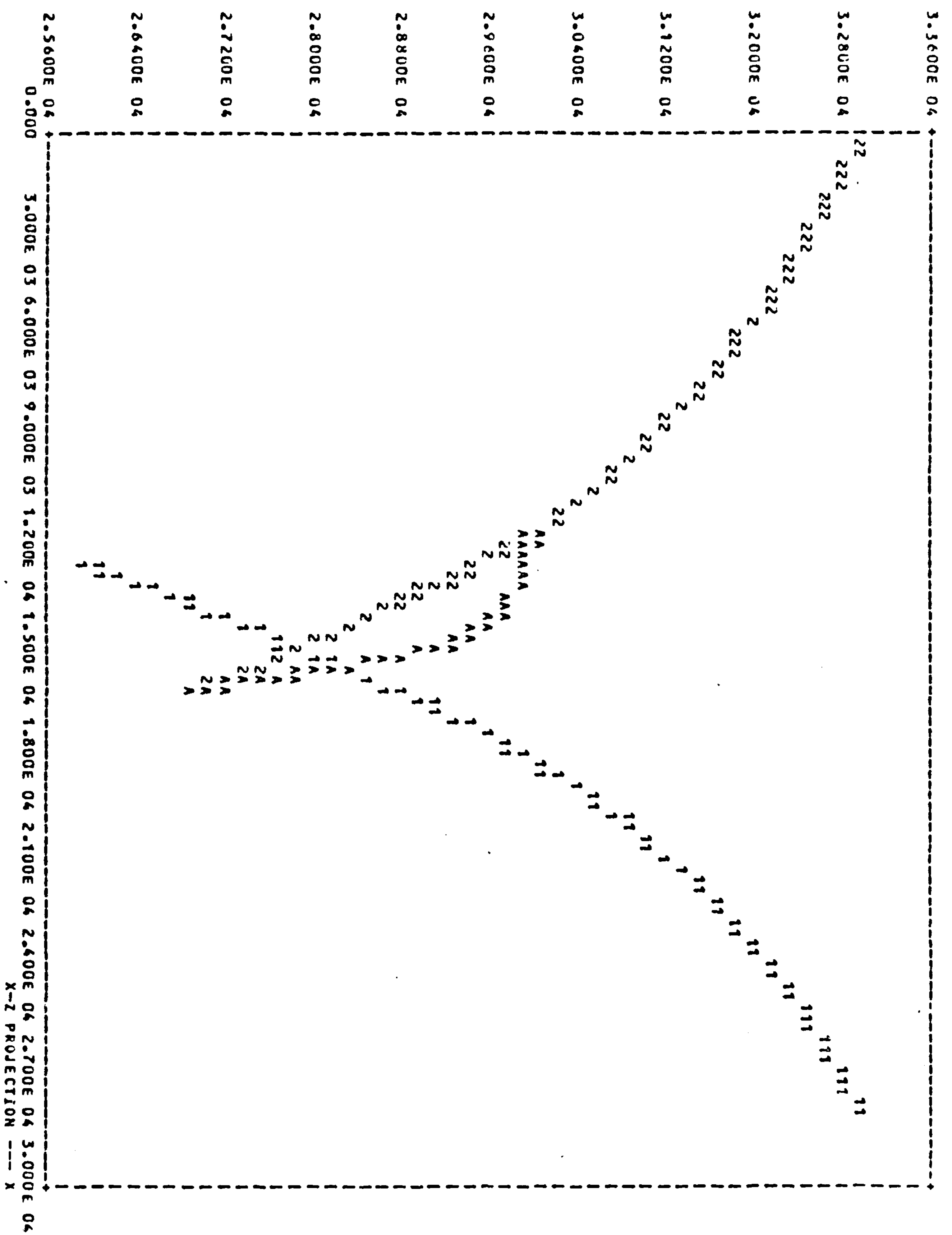


Figure 4-7c
Algorithm 4, Scenario 4



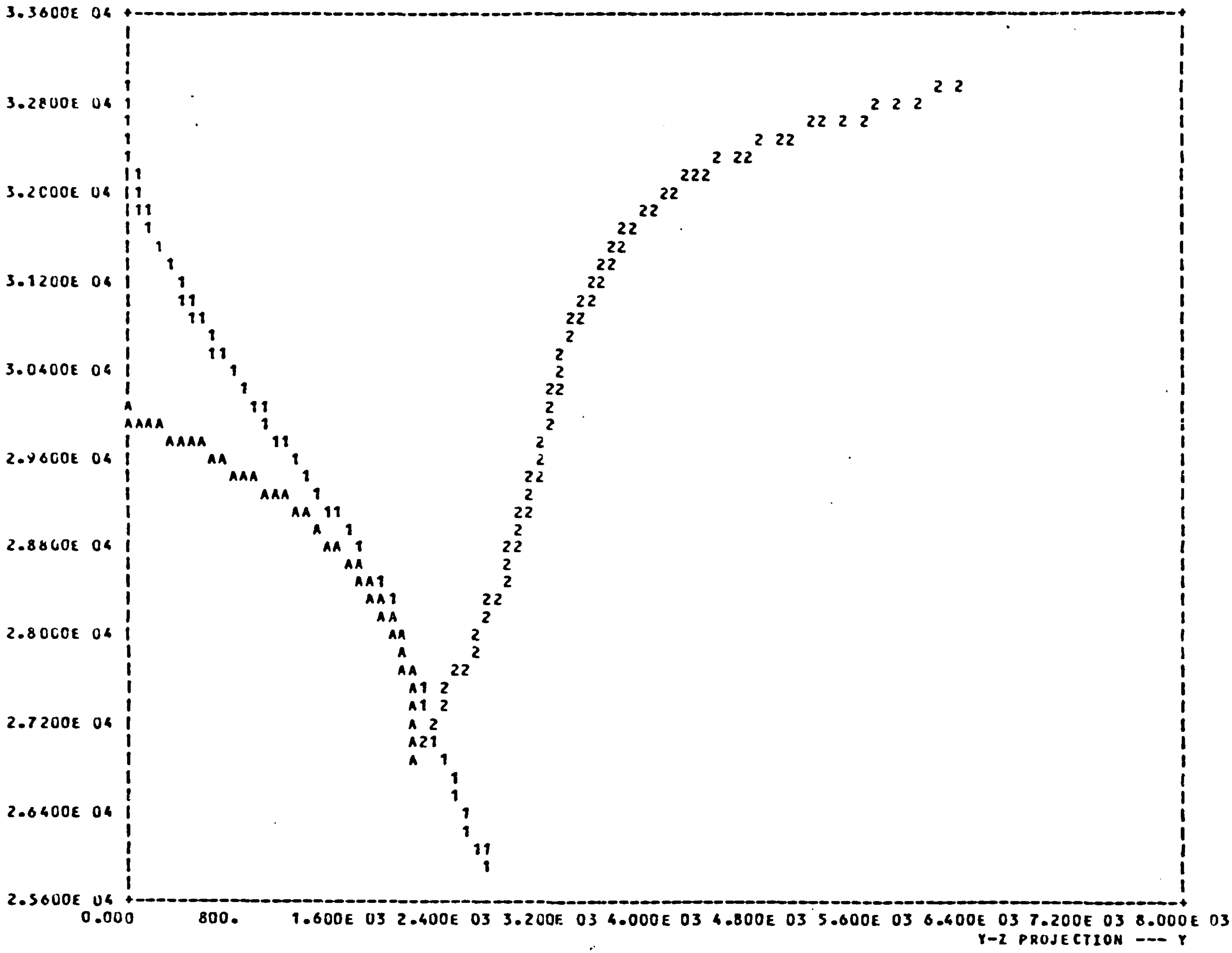


Figure 4-7d
Algorithm 4, Scenario 4

Table 4-8
Algorithm 1, Scenario 5

```
TSTEP =    0.1000, TAU =    0.1500

INIT XO(1) =      15000.0     INIT XO( 7) =      3000.00     INIT XO(13) =      15000.0
INIT XO(2) =     0.000000     INIT XO( 8) =     0.000000     INIT XO(14) =      12000.0
INIT XO(3) =      30000.0     INIT XO( 9) =      30000.0     INIT XO(15) =      30000.0
INIT XO(4) =      1100.00     INIT XO(10) =      3300.00     INIT XO(16) =      3300.00
INIT XO(5) =     0.000000     INIT XO(11) =     0.000000     INIT XO(17) =     0.000000
INIT XO(6) =     0.000000     INIT XO(12) =     0.000000     INIT XO(18) =     -60.0000
                    PROPORTIONAL NAVIGATION GAINS:
          PITCH (RK11) =      4.50     , YAW (RK12) =      4.50

          A/C MAXIMUM LOAD FACTOR =  8.00

           IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =       0.000  DSEP1 =     0.120E 05  DSEP2 =     0.120E 05
START MANEUVER AT T =    0.10
   TIME =       1.000  DSEP1 =     0.989E 04  DSEP2 =     0.919E 04
   TIME =       2.000  DSEP1 =     0.794E 04  DSEP2 =     0.660E 04
   TIME =       3.000  DSEP1 =     0.613E 04  DSEP2 =     0.426E 04
   TIME =       4.000  DSEP1 =     0.449E 04  DSEP2 =     0.206E 04
   TIME =       4.927  DSEP1 =     0.306E 04  DSEP2 =      108.
   TIME =       5.246  DSEP1 =     0.261E 04  DSEP2 =      577.
   TIME =       6.246  DSEP1 =     0.116E 04  DSEP2 =     0.266E 04
   TIME =       7.095  DSEP1 =      35.6      DSEP2 =     0.411E 04
 **** CLOSURE RATE NEGATIVE AT TIME =        7.119****
   TA1 : BEST DSEP =      28.7695     , NOW =      30.8579
   TA2 : BEST DSEP =      52.8064     , NOW =      4150.34

XO(1):       0.2145E 05      XO( 7):    0.2146E 05      XO(13):     0.2173E 05
XO(2):       -660.7          XO( 8):    -832.5          XO(14):     -4955.
XO(3):       0.2870E 05      XO( 9):    0.2869E 05      XO(15):     0.2932E 05
XO(4):        753.9          XO(10):     2041.          XO(16):      1965.
XO(5):       -26.19          XO(11):    -10.97          XO(17):     -2.076
XO(6):       -13.55          XO(12):    -4.750          XO(18):     -52.37

        DELX1:     -11.5      DELY1:      -28.3
        DELZ1:      4.48      DMIS1:       30.9
        BEST DMIS WAS      28.8

        DELX2:     -276.      DELY2:     0.409E 04
        DELZ2:     -621.      DMIS2:     0.415E 04
        BEST DMIS WAS      52.8
```

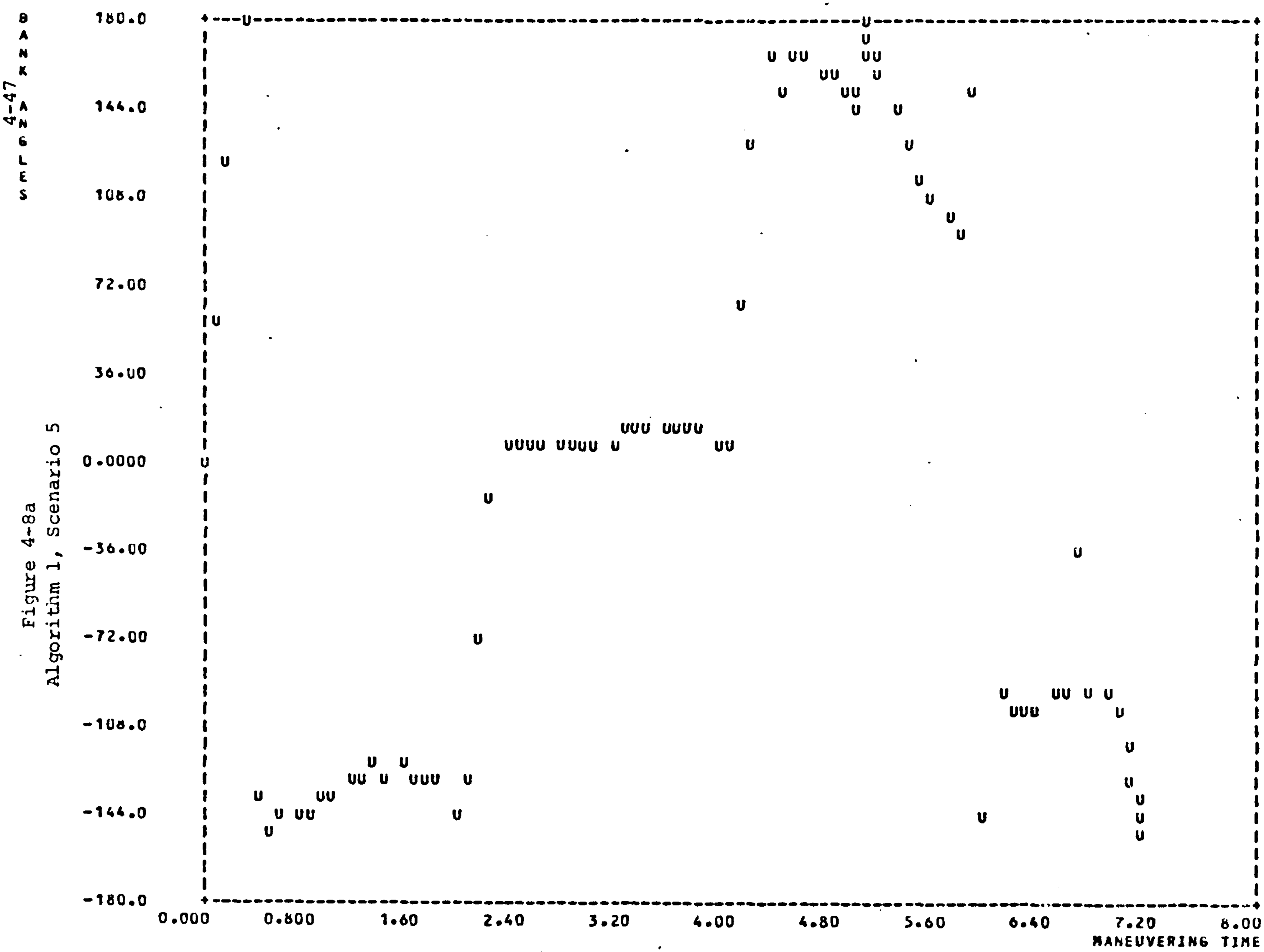


Figure 4-8a
Algorithm 1, Scenario 5

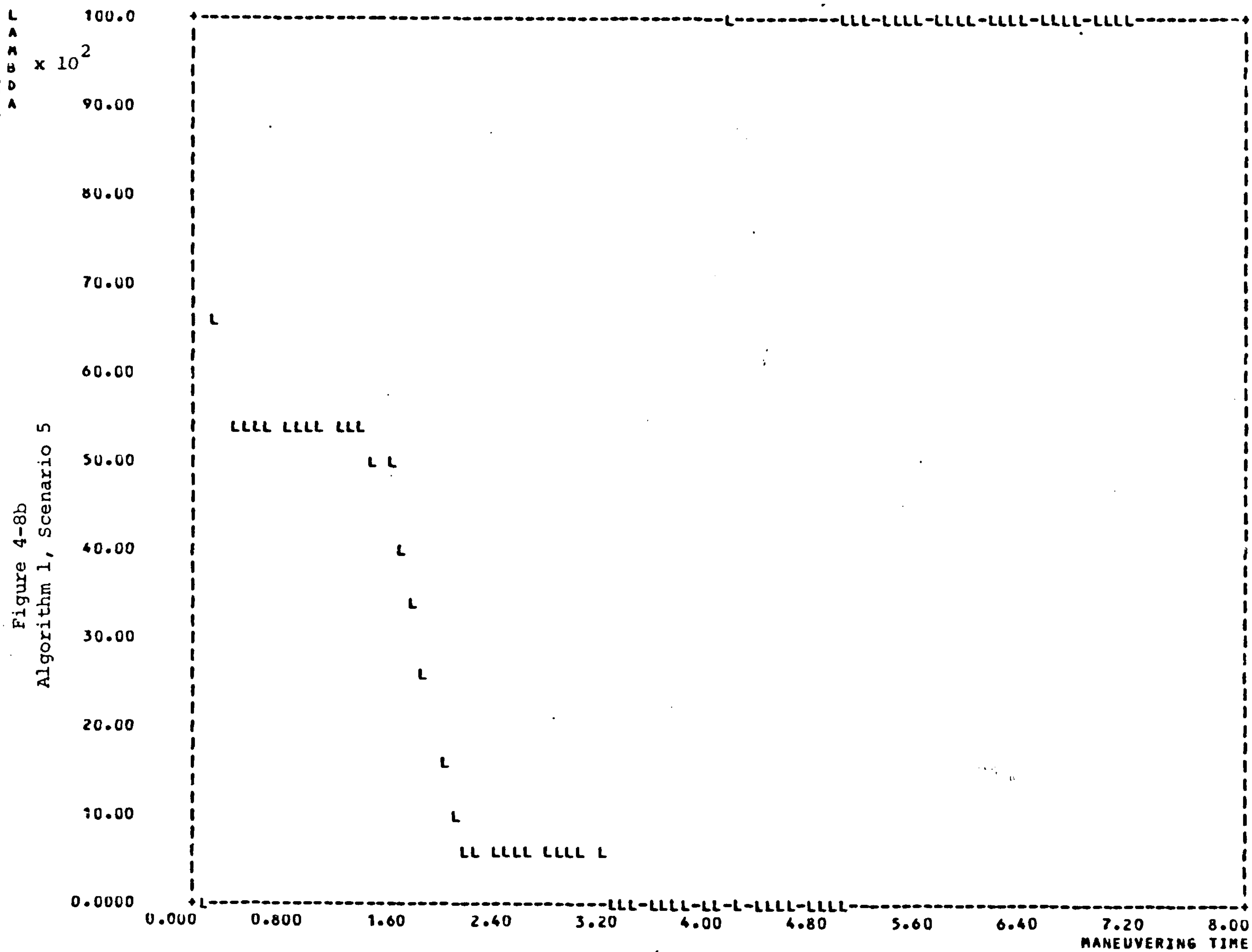


Figure 4-8b
Algorithm 1, Scenario 5

Figure 4-8c
Algorithm 1, Scenario 5

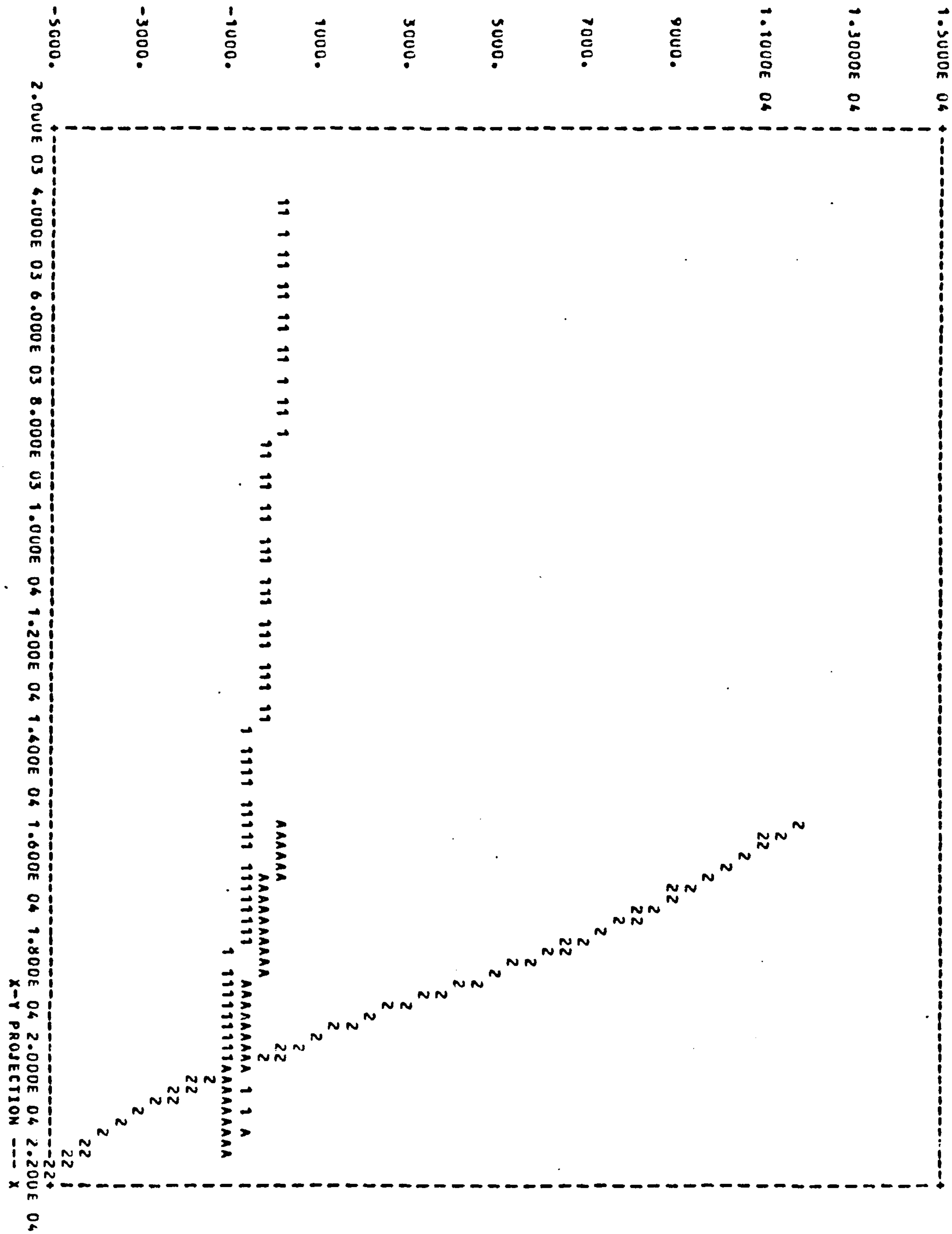
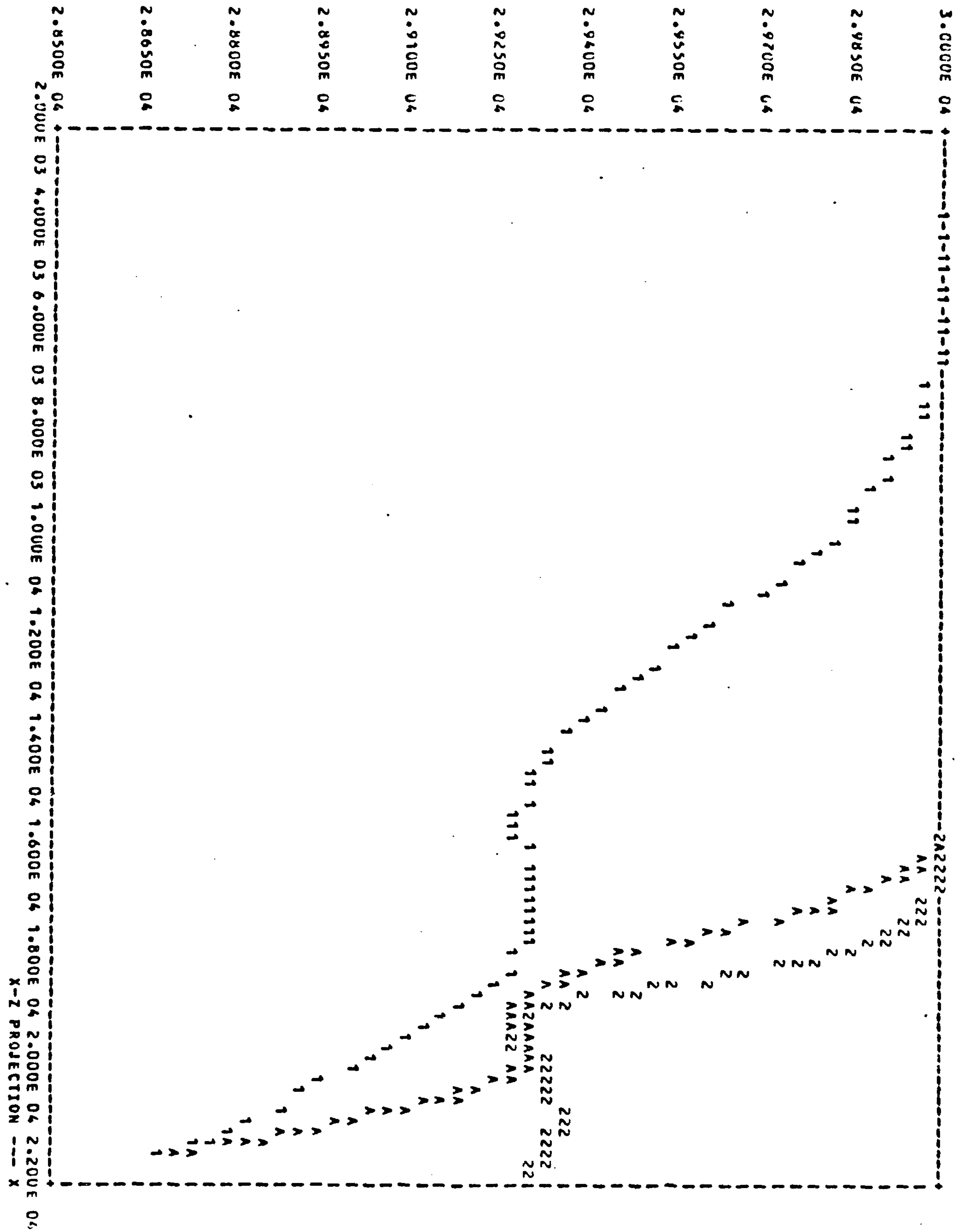


Figure 4-8d
Algorithm 1, Scenario 5



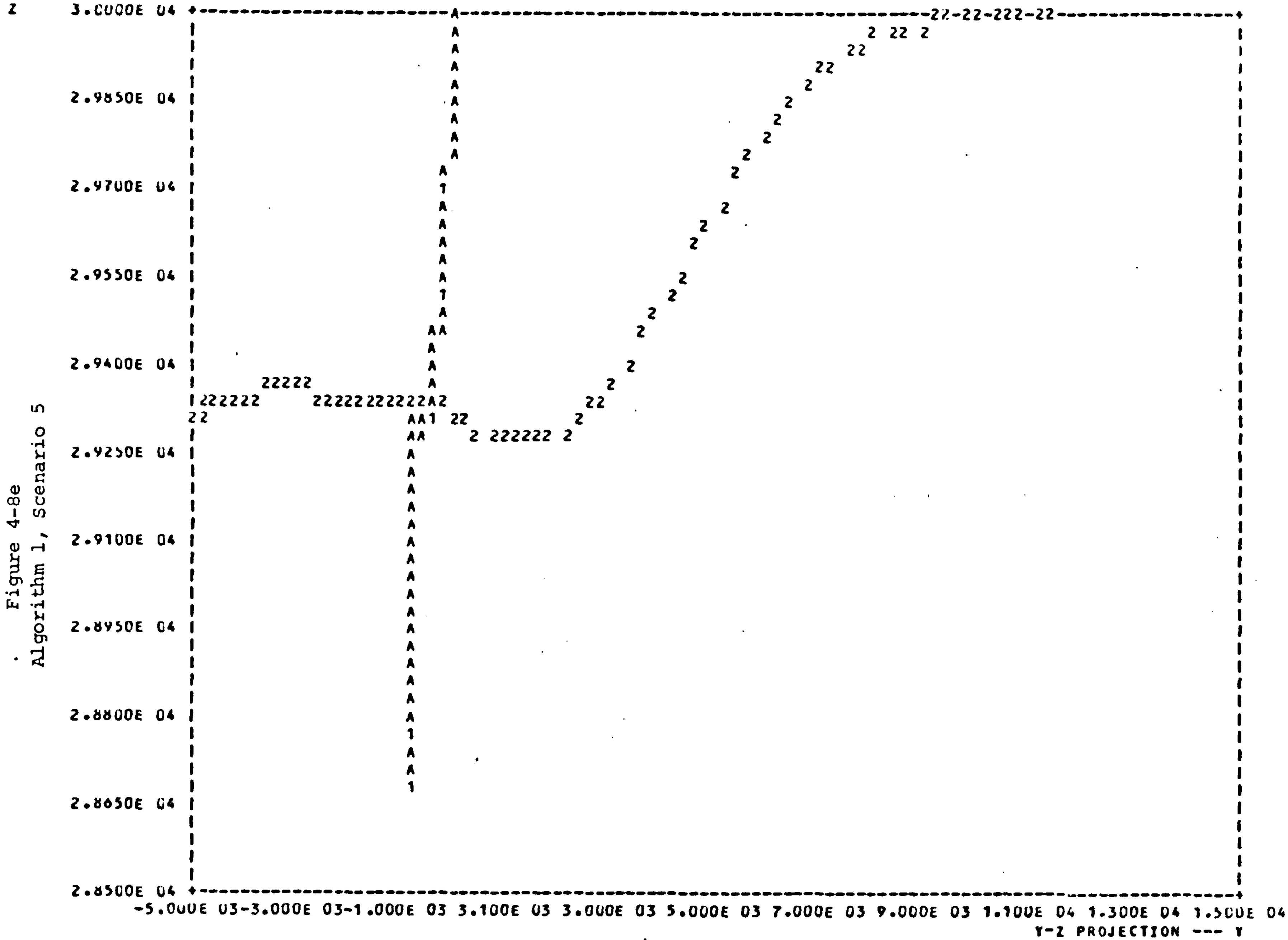


Figure 4-8e
Algorithm 1, Scenario 5

Table 4-9
Algorithm 2, Scenario 5

```
TSTEP =   0.1000, TAU =   0.1500

INIT XO(1) =      15000.0     INIT XO( 7) =     3000.00     INIT XO(13) =     15000.0
INIT XO(2) =     0.000000     INIT XO( 8) =    0.000000     INIT XO(14) =     12000.0
INIT XO(3) =      30000.0     INIT XO( 9) =     30000.0     INIT XO(15) =     30000.0
INIT XO(4) =      1100.00     INIT XO(10) =     3300.00     INIT XO(16) =     3300.00
INIT XO(5) =     0.000000     INIT XO(11) =    0.000000     INIT XO(17) =    0.000000
INIT XO(6) =     0.000000     INIT XO(12) =    0.000000     INIT XO(18) =    -60.0000
                  PROPORTIONAL NAVIGATION GAINS:
        PITCH (RK11) =     4.50    , YAW (RK12) =     4.50

        A/C MAXIMUM LOAD FACTOR =  8.00

         IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =      0.000  DSEP1 =    0.120E 05  DSEP2 =    0.120E 05
START MANEUVER AT T =   0.10
   TIME =      1.000  DSEP1 =    0.989E 04  DSEP2 =    0.914E 04
   TIME =      2.000  DSEP1 =    0.794E 04  DSEP2 =    0.637E 04
   TIME =      3.000  DSEP1 =    0.613E 04  DSEP2 =    0.368E 04
   TIME =      4.000  DSEP1 =    0.445E 04  DSEP2 =    0.106E 04
   TIME =      4.473  DSEP1 =    0.368E 04  DSEP2 =     130.
   TIME =      5.311  DSEP1 =    0.234E 04  DSEP2 =    0.210E 04
   TIME =      6.311  DSEP1 =     825.      DSEP2 =    0.418E 04
   TIME =      6.683  DSEP1 =     19.5      DSEP2 =    0.525E 04
 **** CLOSURE RATE NEGATIVE AT TIME =     6.683****
   TA1 : BEST DSEP =     17.1382    , NOW =     19.4535
   TA2 : BEST DSEP =     51.6636    , NOW =     5254.68

XO(1):   0.2091E 05     XO( 7):    0.2092E 05    XO(13):    0.2037E 05
XO(2):    396.3         XO( 8):     412.6        XO(14):    -4824.
XO(3):   0.2959E 05     XO( 9):    0.2959E 05    XO(15):    0.2984E 05
XO(4):    731.5         XO(10):     2071.        XO(16):     2043.
XO(5):   -30.91         XO(11):    -12.44        XO(17):    -8.208
XO(6):   -24.54         XO(12):    -10.27        XO(18):    -86.82

       DELX1:    -10.2       DELY1:    -16.2
       DELZ1:     3.35       DMIS1:     19.5
       BEST DMIS WAS     17.1

       DELX2:     543.       DELY2:    0.522E 04
       DELZ2:    -252.       DMIS2:    0.525E 04
       BEST DMIS WAS     51.7
```

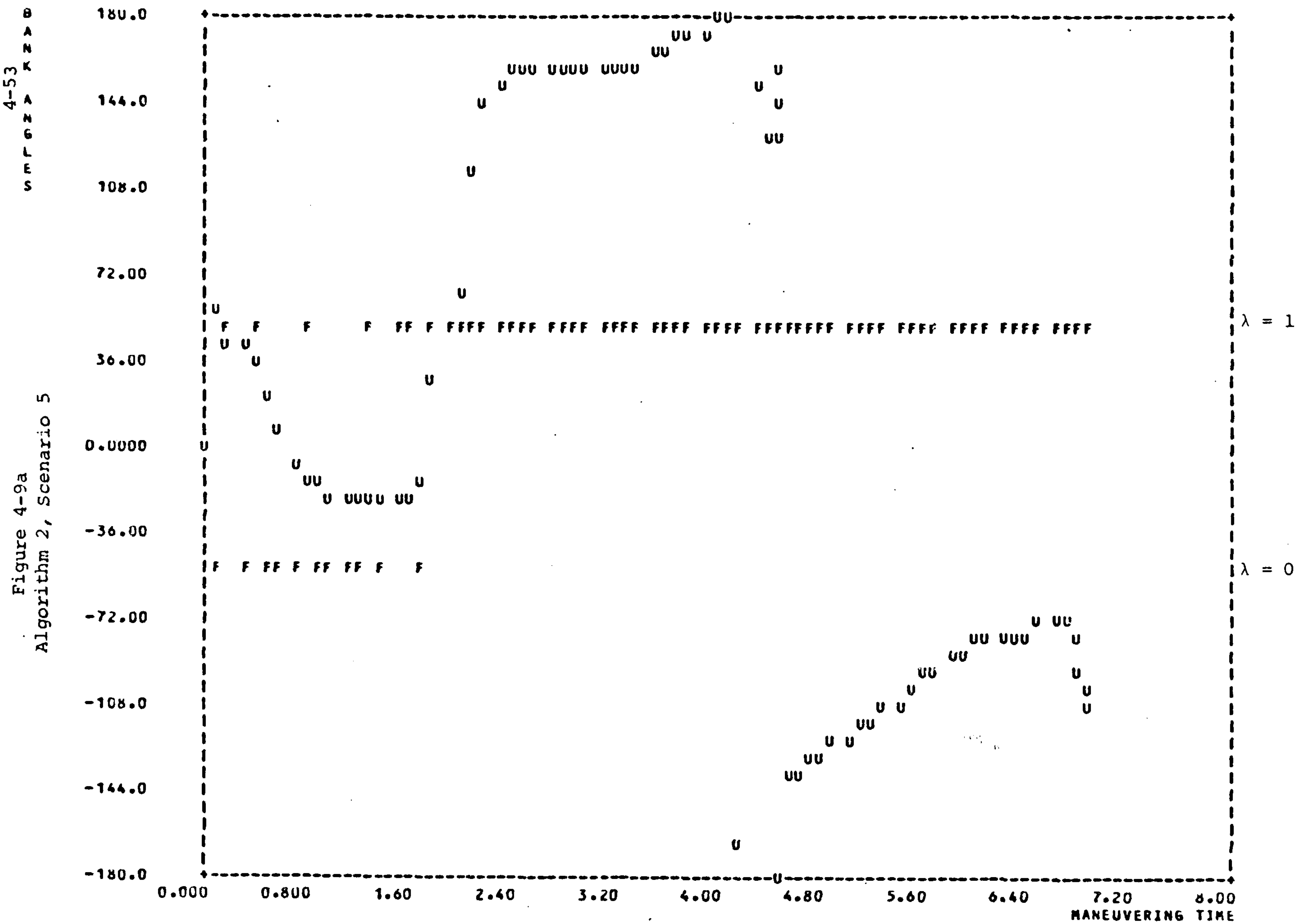


Figure 4-9a
Algorithm 2, Scenario 5

Figure 4-9b
Algorithm 2, Scenario 5

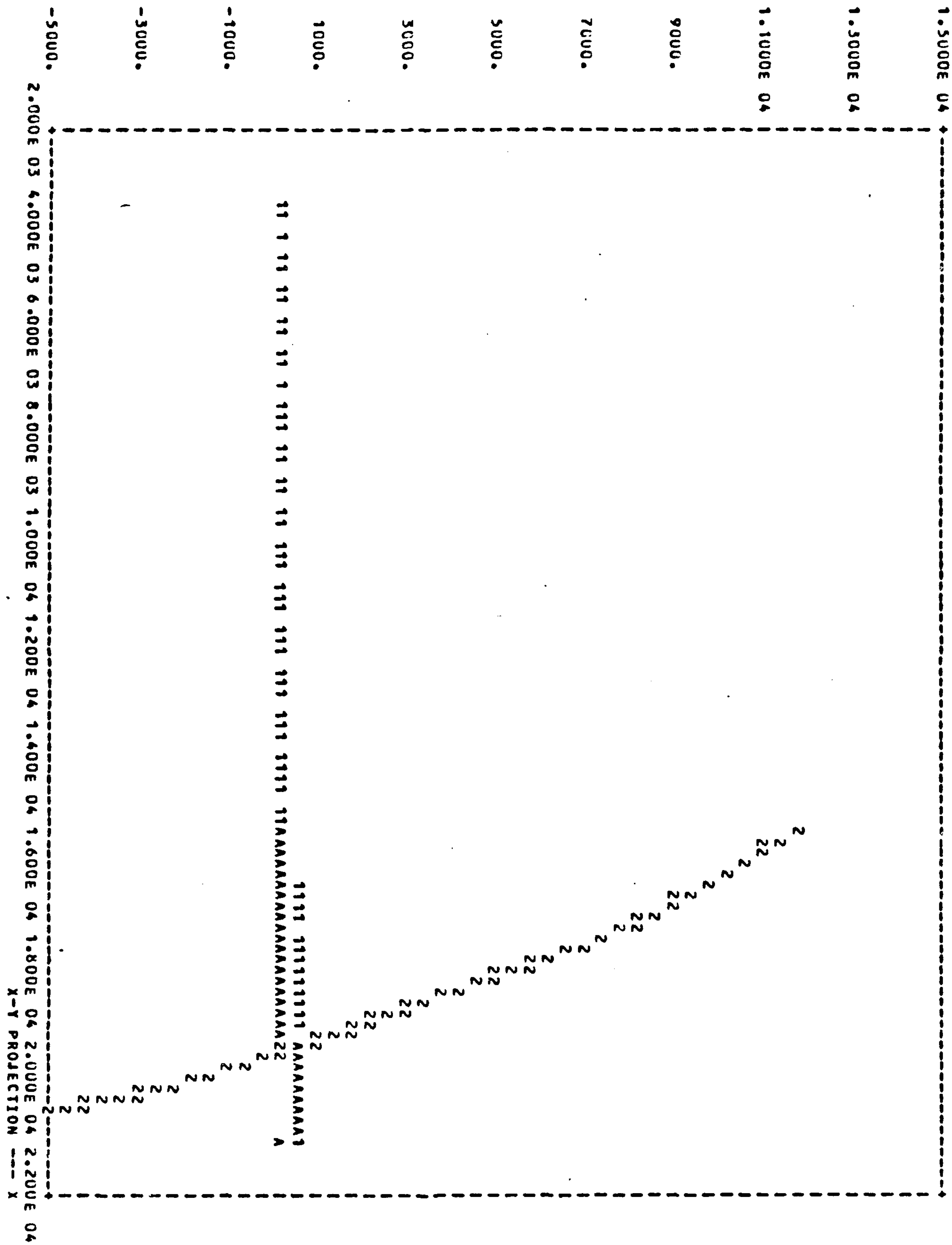


Figure 4-9c
Algorithm 2, Scenario 5

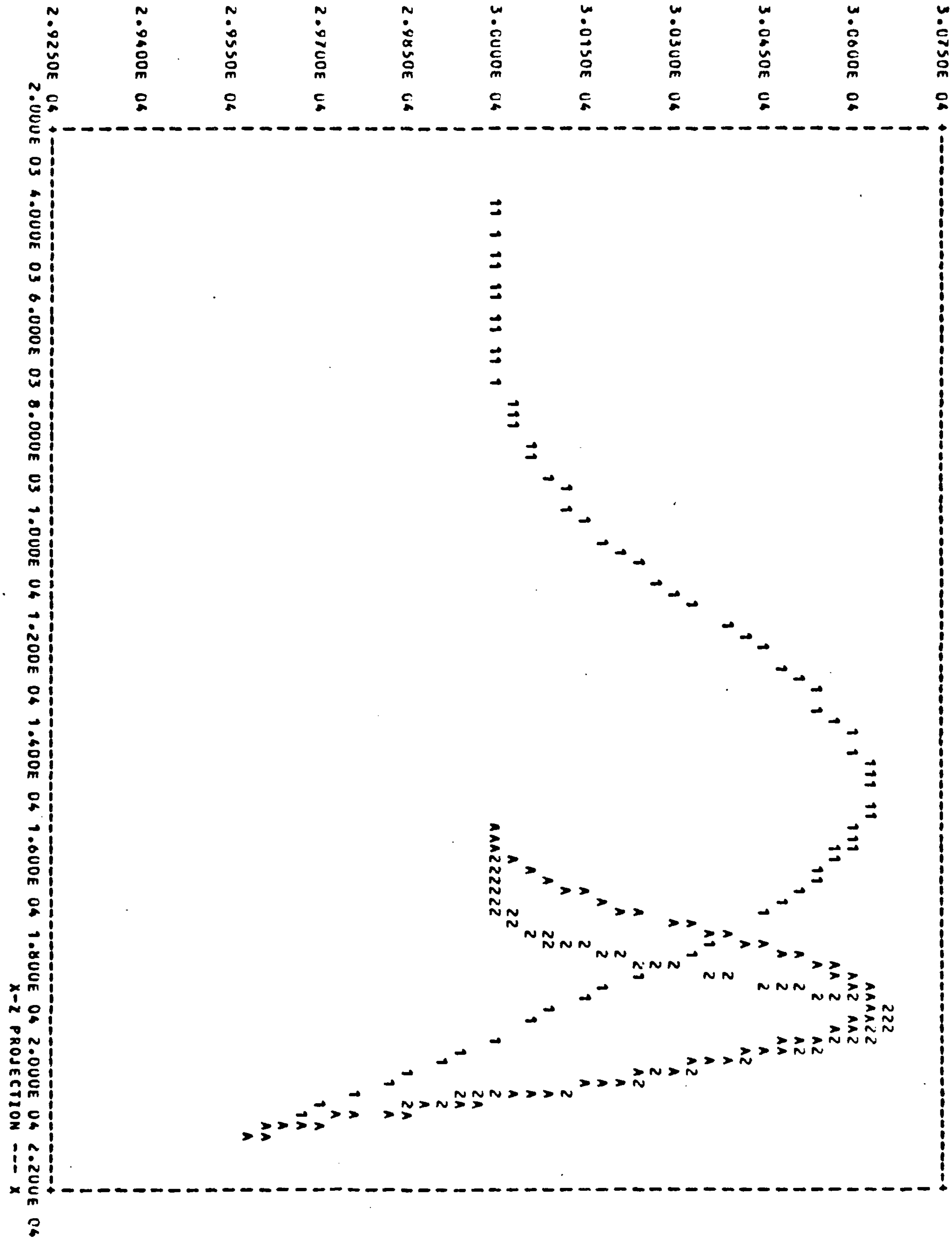


Figure 4-9d
Algorithm 2, Scenario 5

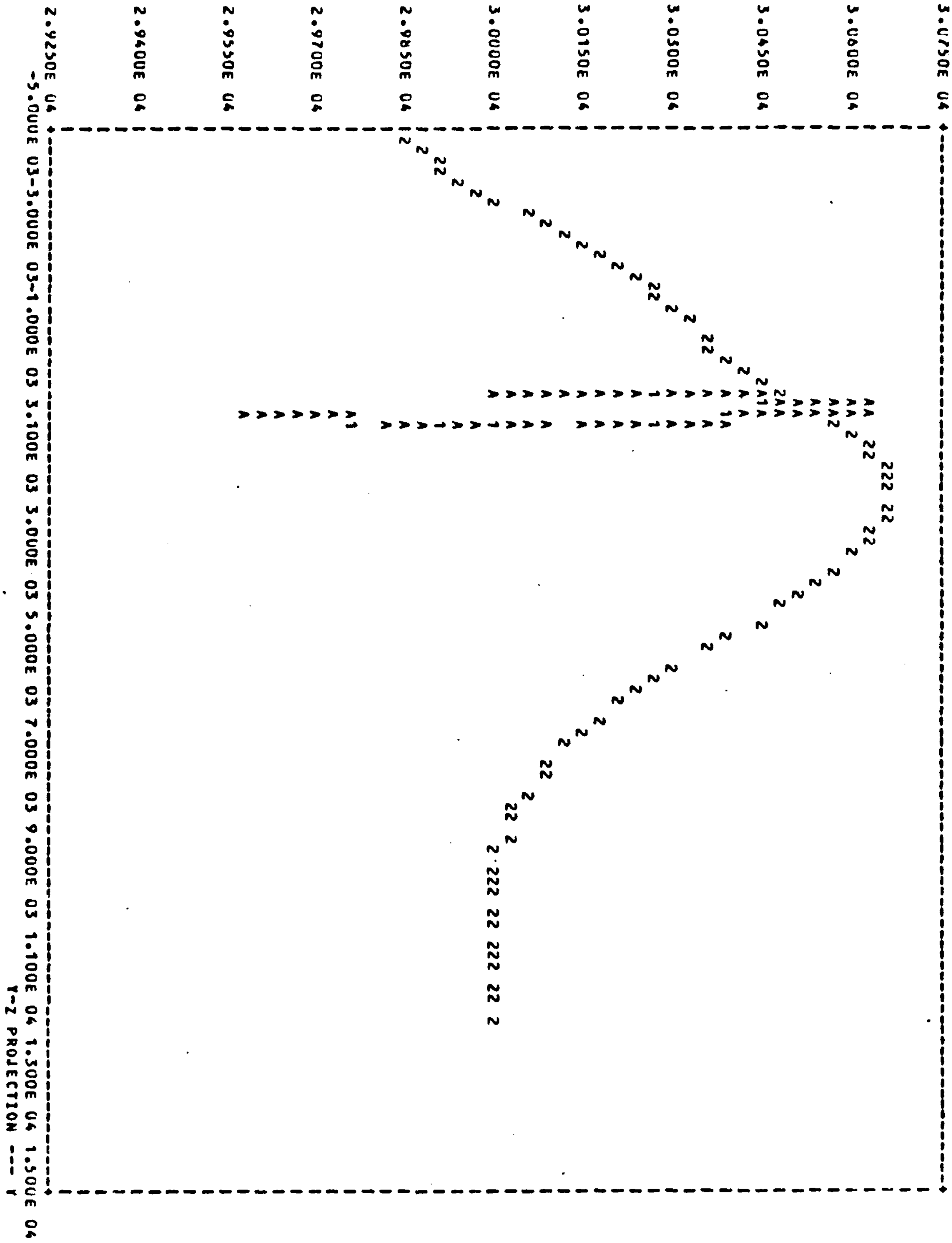


Table 4-10
Algorithm 1, Scenario 6

```
TSTEP =   0.1000, TAU =   0.1500

INIT X0(1) =    13000.0       INIT X0( 7) =    30000.0      INIT X0(13) =   0.000000
INIT X0(2) =    0.000000      INIT X0( 8) =    0.000000     INIT X0(14) =   0.000000
INIT X0(3) =    [illegible]   INIT X0( 9) =    33000.0      INIT X0(15) =   33000.0
INIT X0(4) =    1100.00       INIT X0(10) =    3300.00      INIT X0(16) =   3300.00
INIT X0(5) =    0.000000      INIT X0(11) =    -7.99999     INIT X0(17) =   -7.99999
INIT X0(6) =    0.000000      INIT X0(12) =    180.000      INIT X0(18) =   0.000000
                    PROPORTIONAL NAVIGATION GAINS:
        PITCH (RK11) =      4.50    , YAW (RK12) =     4.50

        A/C MAXIMUM LOAD FACTOR =  8.00

         IN MANEUVER, AFTERBURNERS WILL BE ON

   TIME =        0.000  DSEP1 =    0.173E 05  DSEP2 =    0.133E 05
START MANEUVER AT T =    0.10
   TIME =        1.000  DSEP1 =    0.130E 05  DSEP2 =    0.112E 05
   TIME =        2.000  DSEP1 =    0.903E 04  DSEP2 =    0.926E 04
   TIME =        3.000  DSEP1 =    0.531E 04  DSEP2 =    0.743E 04
   TIME =        4.000  DSEP1 =    0.177E 04  DSEP2 =    0.572E 04
   TIME =        4.539  DSEP1 =     72.9      DSEP2 =    0.453E 04
   TIME =        5.138  DSEP1 =    0.197E 04  DSEP2 =    0.385E 04
   TIME =        6.138  DSEP1 =    0.487E 04  DSEP2 =    0.236E 04
   TIME =        7.138  DSEP1 =    0.749E 04  DSEP2 =     925.
   TIME =        7.618  DSEP1 =    0.906E 04  DSEP2 =     22.1
 **** CLOSURE RATE NEGATIVE AT TIME =        7.618****
   TA1 : BEST DSEP =     55.6756     , NOW =      9072.77
   TA2 : BEST DSEP =     22.1066     , NOW =      24.5693

X0(1):      0.1993E 05      X0( 7):     0.1120E 05     X0(13):     0.1994E 05
X0(2):       15.83          X0( 8):      2086.         X0(14):     -5.526
X0(3):      0.268[illegible]E 05  X0( 9):     0.2745E 05     X0(15):     0.2680E 05
X0(4):       713.6          X0(10):      1935.         X0(16):      2033.
X0(5):      -28.21          X0(11):     -19.64         X0(17):     -20.18
X0(6):       16.04          X0(12):      137.7         X0(18):      7.204

      DELX1:      0.873E 04   DELY1:     -0.207E 04
      DELZ1:      0.135E 04   DMIS1:      0.907E 04
      BEST DMIS WAS      55.7

      DELX2:      -12.1       DELY2:      21.4
      DELZ2:      0.598       DMIS2:      24.6
      BEST DMIS WAS      22.1
```

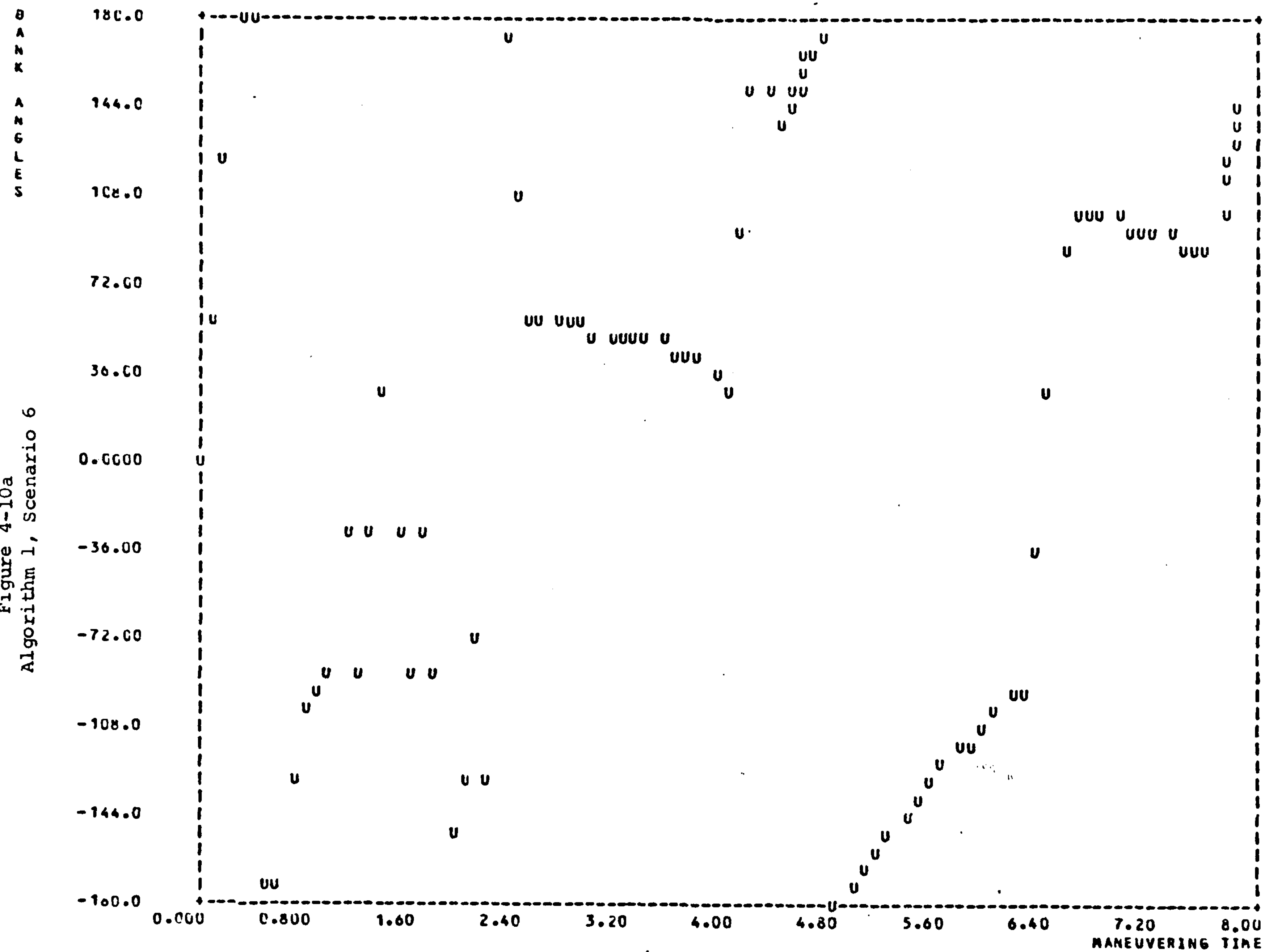


Figure 4-10a
Algorithm 1, Scenario 6

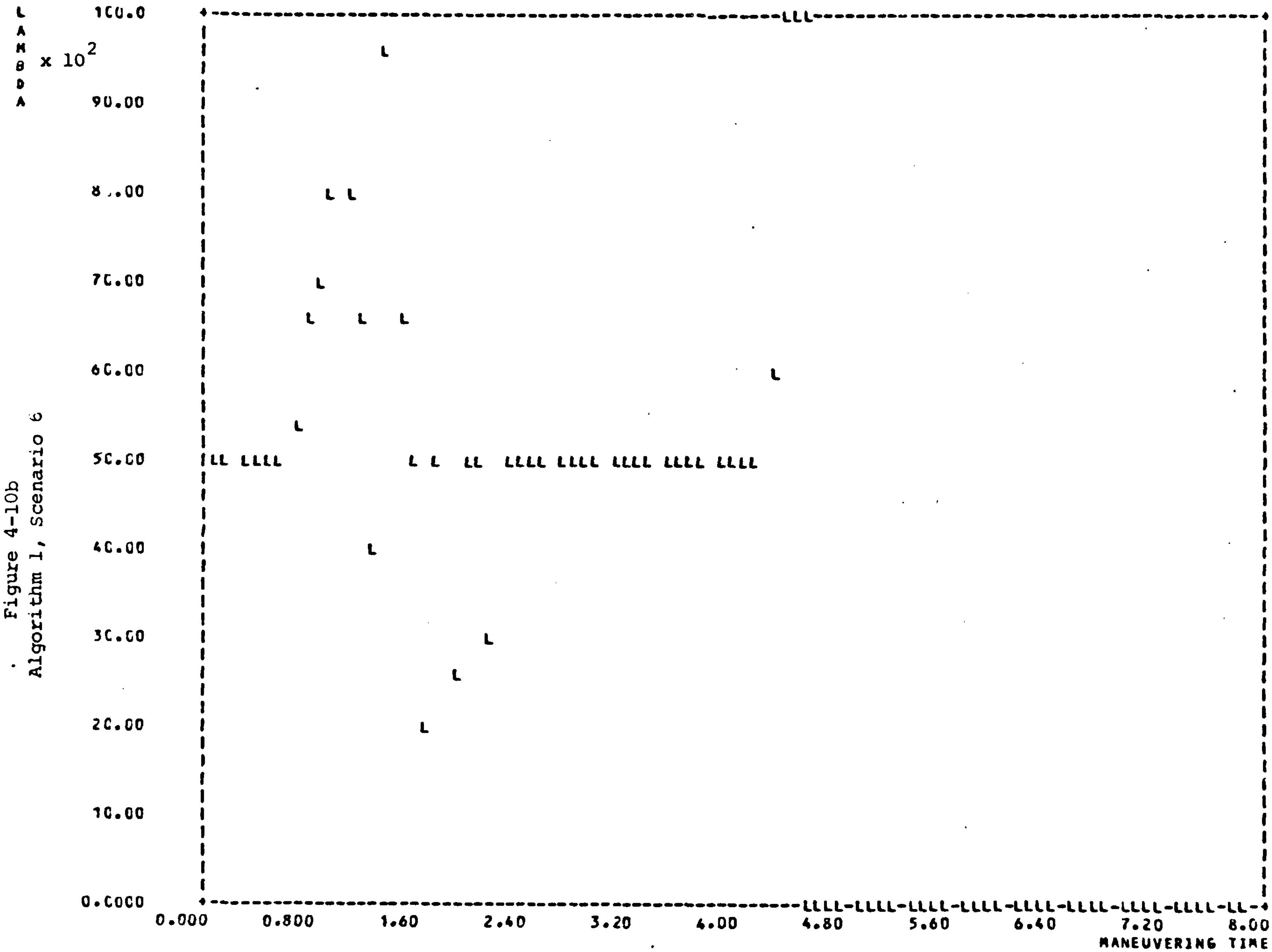


Figure 4-10b
Algorithm 1, Scenario 6

Figure 4-10c
Algorithm 1, Scenario 6

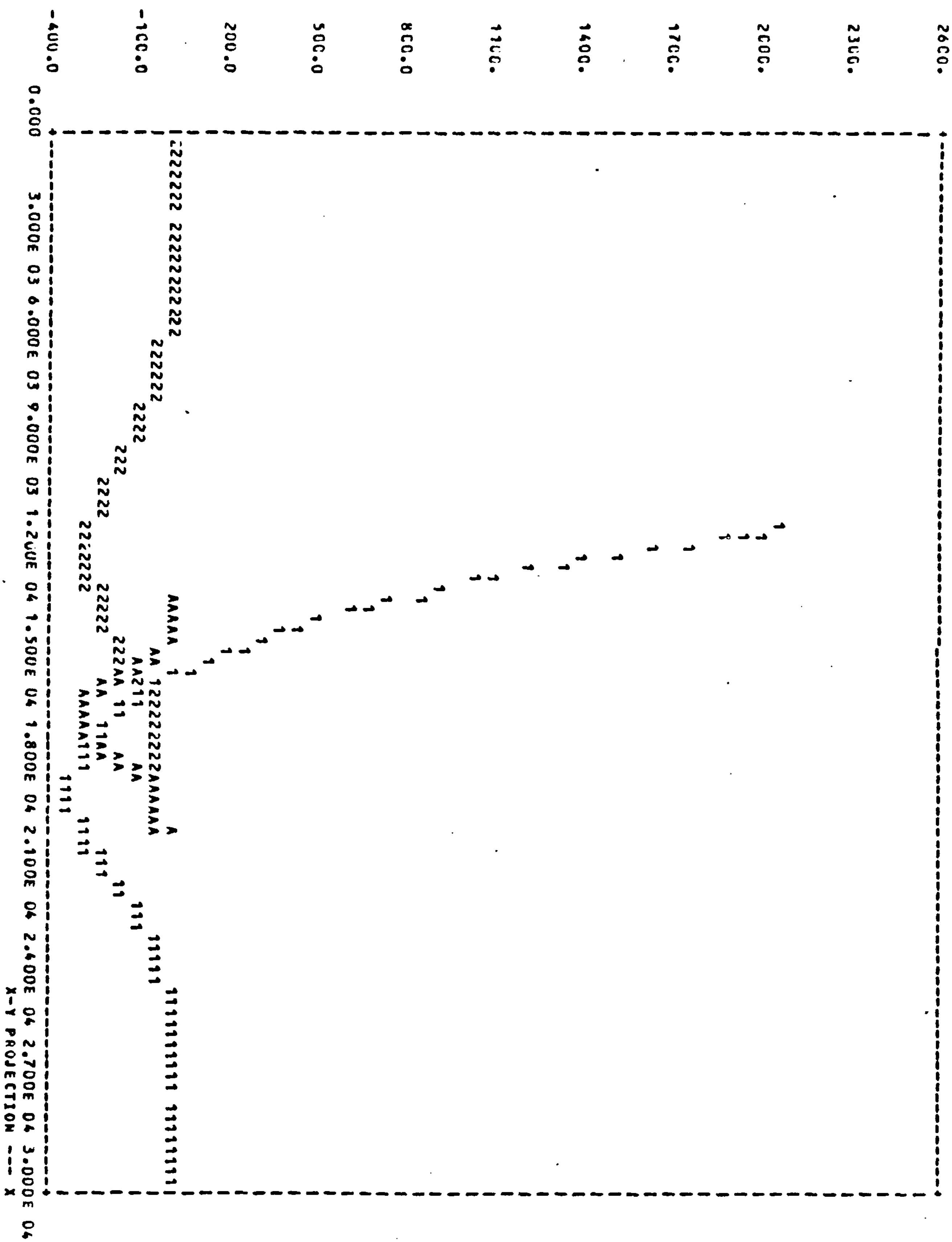


Figure 4-10d
Algorithm 1, Scenario 6

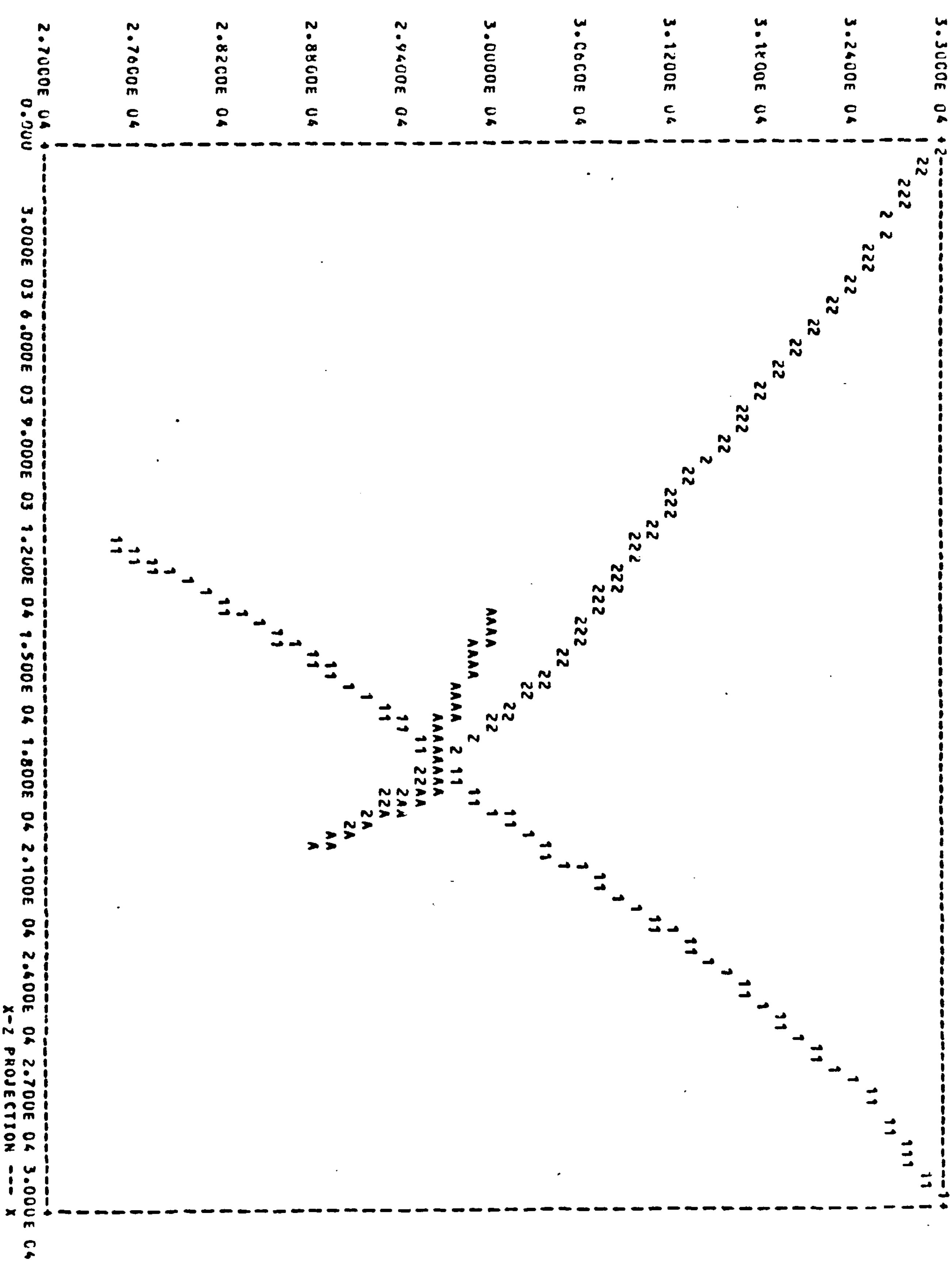


Figure 4-10e
Algorithm 1, Scenario 6

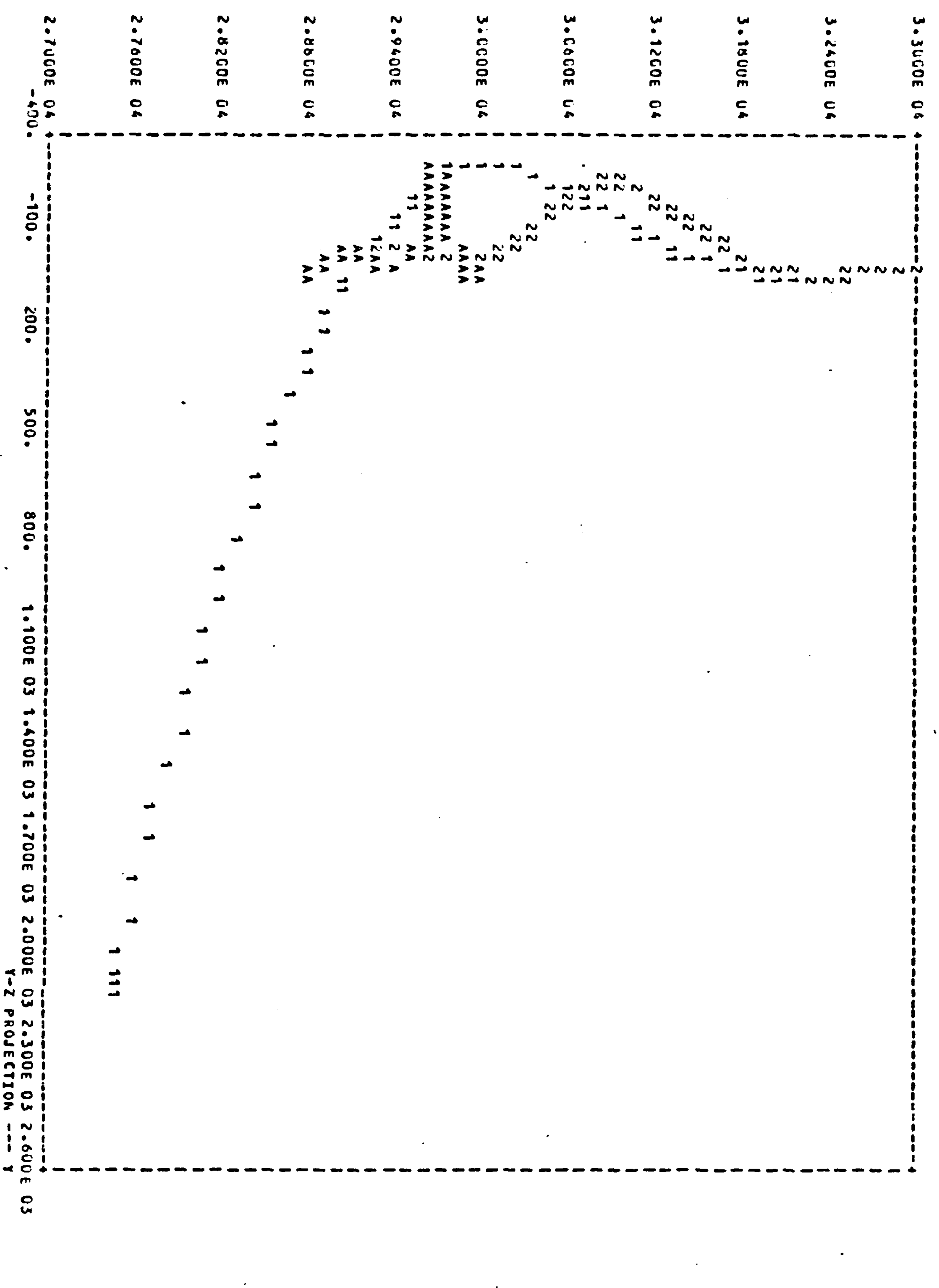
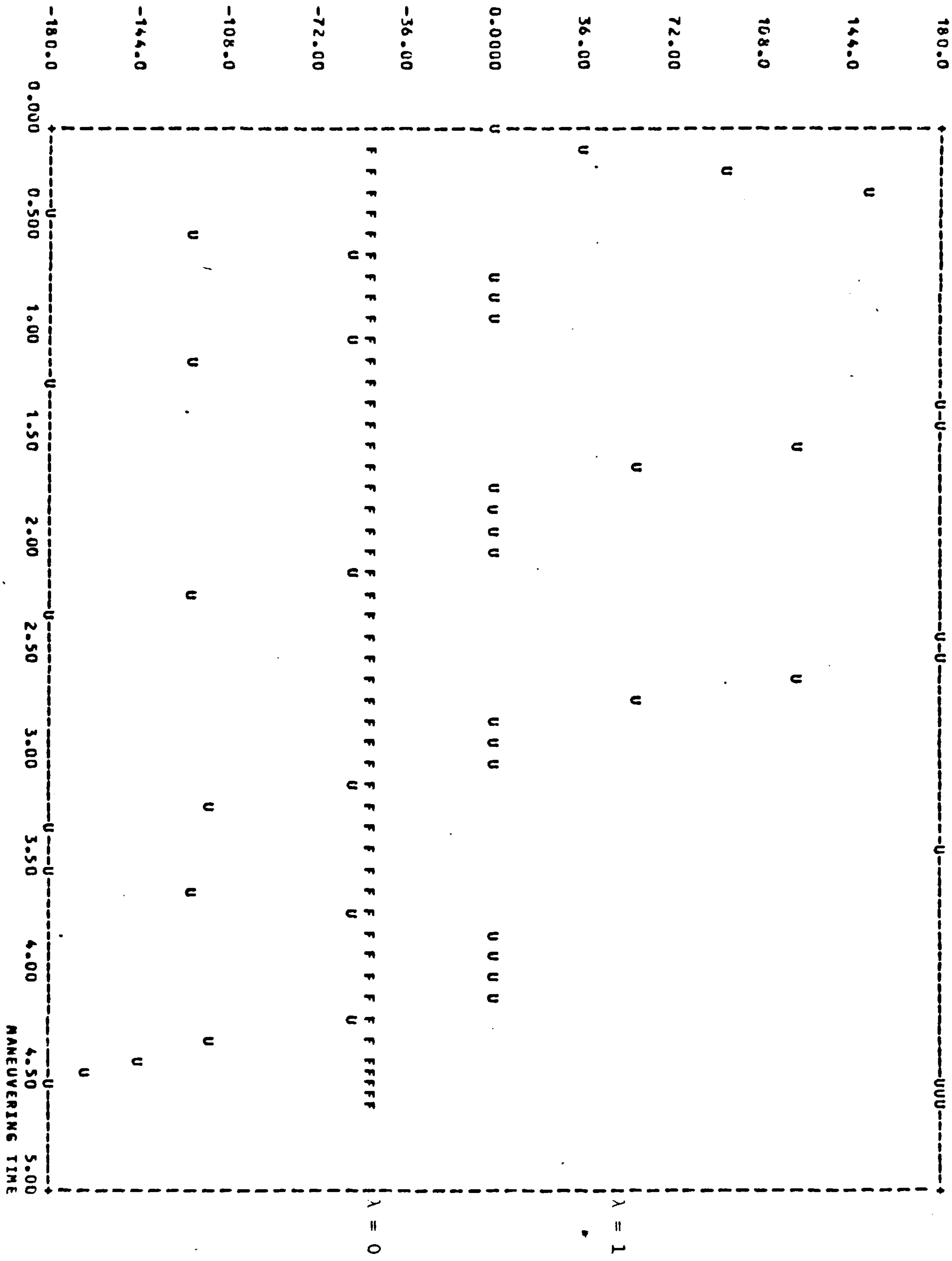


Table 4-11
Algorithm 4, Scenario 7

```
TSTEP =    0.1000, TAU =    0.1500

INIT XO(1) =      0.000000     INIT XO( 7) =      15000.0     INIT XO(13) =     0.000000
INIT XO(2) =      0.000000     INIT XO( 8) =     0.000000     INIT XO(14) =      15000.0
INIT XO(3) =       30000.0     INIT XO( 9) =      30000.0     INIT XO(15) =      30000.0
INIT XO(4) =       1000.00     INIT XO(10) =      3300.00     INIT XO(16) =      3300.00
INIT XO(5) =      0.000000     INIT XO(11) =     0.000000     INIT XO(17) =     0.000000
INIT XO(6) =       45.0000     INIT XO(12) =      180.000     INIT XO(18) =     -89.9998
              PROPORTIONAL NAVIGATION GAINS:
      PITCH (RK11) =      4.50     , YAW (RK12) =      4.50

      A/C MAXIMUM LOAD FACTOR =  8.00

       IN MANEUVER, AFTERBURNERS WILL BE ON

  TIME =       0.000   DSEP1 =    0.150E 05   DSEP2 =     0.150E 05
START MANEUVER AT T =    0.10
  TIME =       1.000   DSEP1 =    0.112E 05   DSEP2 =     0.111E 05
  TIME =       2.000   DSEP1 =    0.760E 04   DSEP2 =     0.759E 04
  TIME =       3.000   DSEP1 =    0.434E 04   DSEP2 =     0.437E 04
  TIME =       4.000   DSEP1 =    0.136E 04   DSEP2 =     0.146E 04
 **** CLOSURE RATE NEGATIVE AT TIME =       4.588****
  TA1 : BEST DSEP =      157.707     , NOW =      341.083
  TA2 : BEST DSEP =      268.416     , NOW =      298.517

XO(1):      2883.          XO( 7):      2552.          XO(13):      2648.
XO(2):      2731.          XO( 8):      2652.          XO(14):      2548.
XO(3):    0.2982E 05       XO( 9):    0.2981E 05       XO(15):    0.2981E 05
XO(4):      725.2          XO(10):      2371.          XO(16):      2370.
XO(5):     -6.204          XO(11):     -2.482          XO(17):     -2.318
XO(6):      36.00          XO(12):      152.3          XO(18):     -61.79

      DELX1:      332.       DELY1:       79.2
      DELZ1:      9.56       DMIS1:       341.
      BEST DMIS WAS       158.

      DELX2:      236.       DELY2:       183.
      DELZ2:      6.00       DMIS2:       299.
      BEST DMIS WAS       268.
```

Figure 4-11a
Algorithm 4, Scenario 7



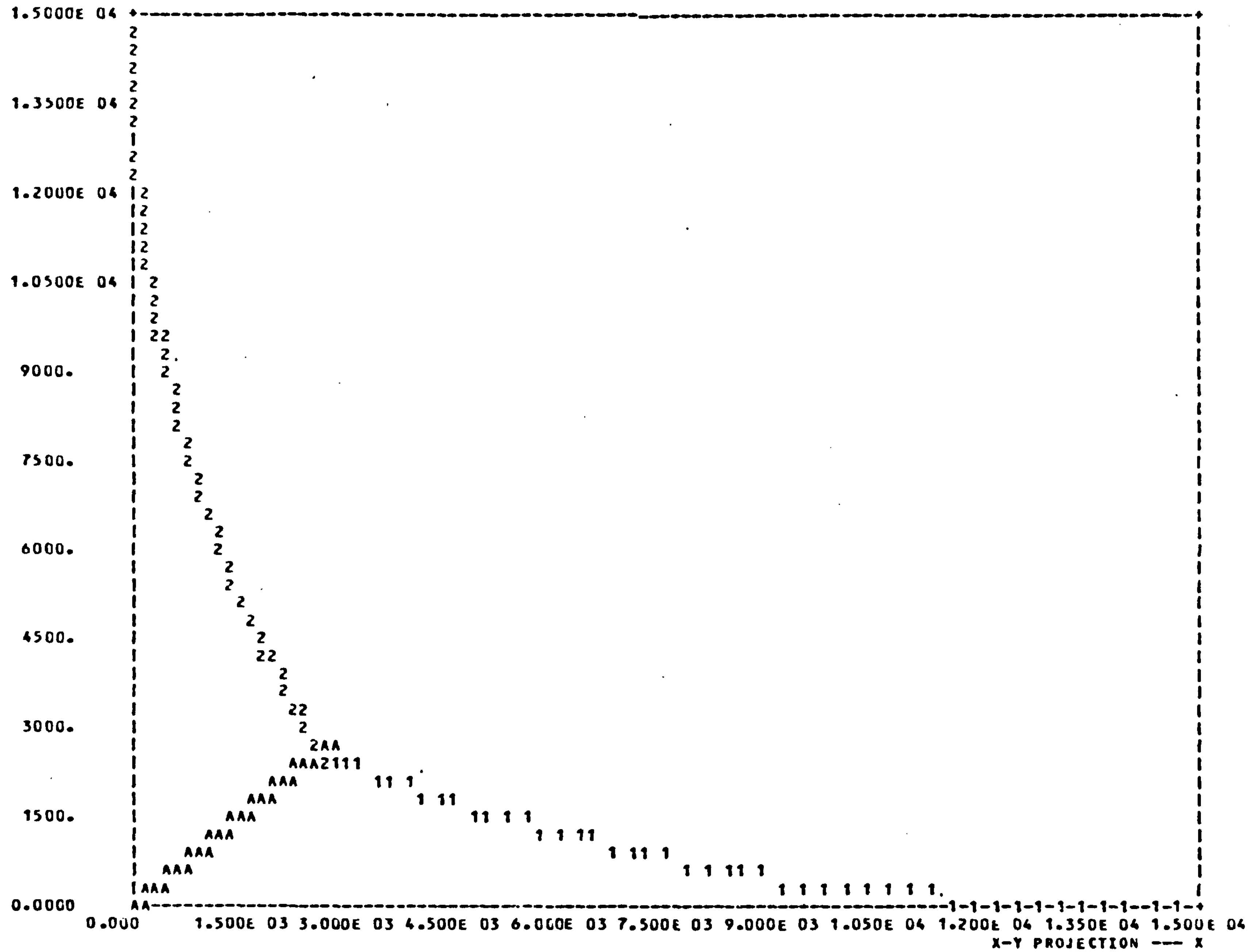
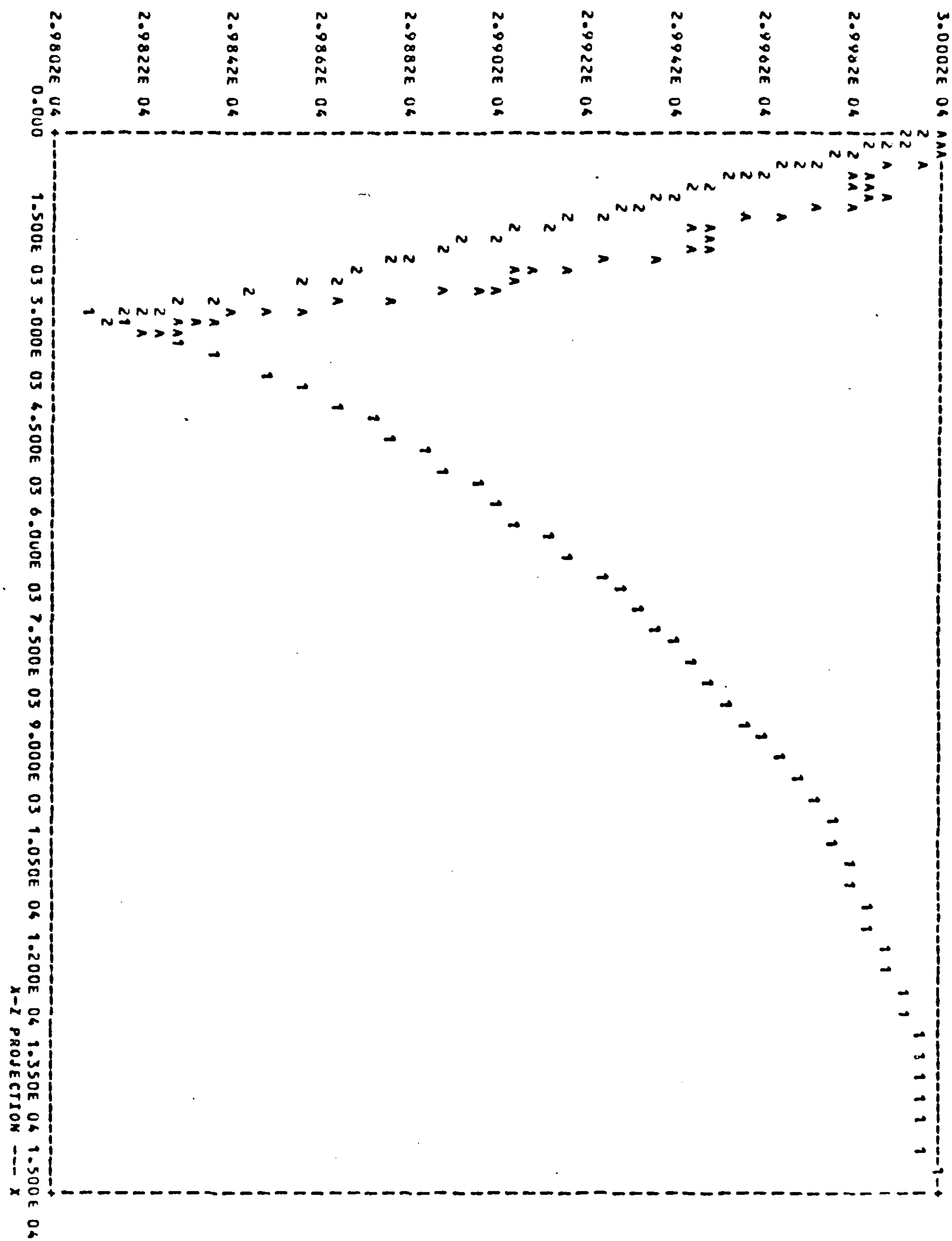


Figure 4-11b
Algorithm 4, Scenario 7

Figure 4-11c
Algorithm 4, Scenario 7



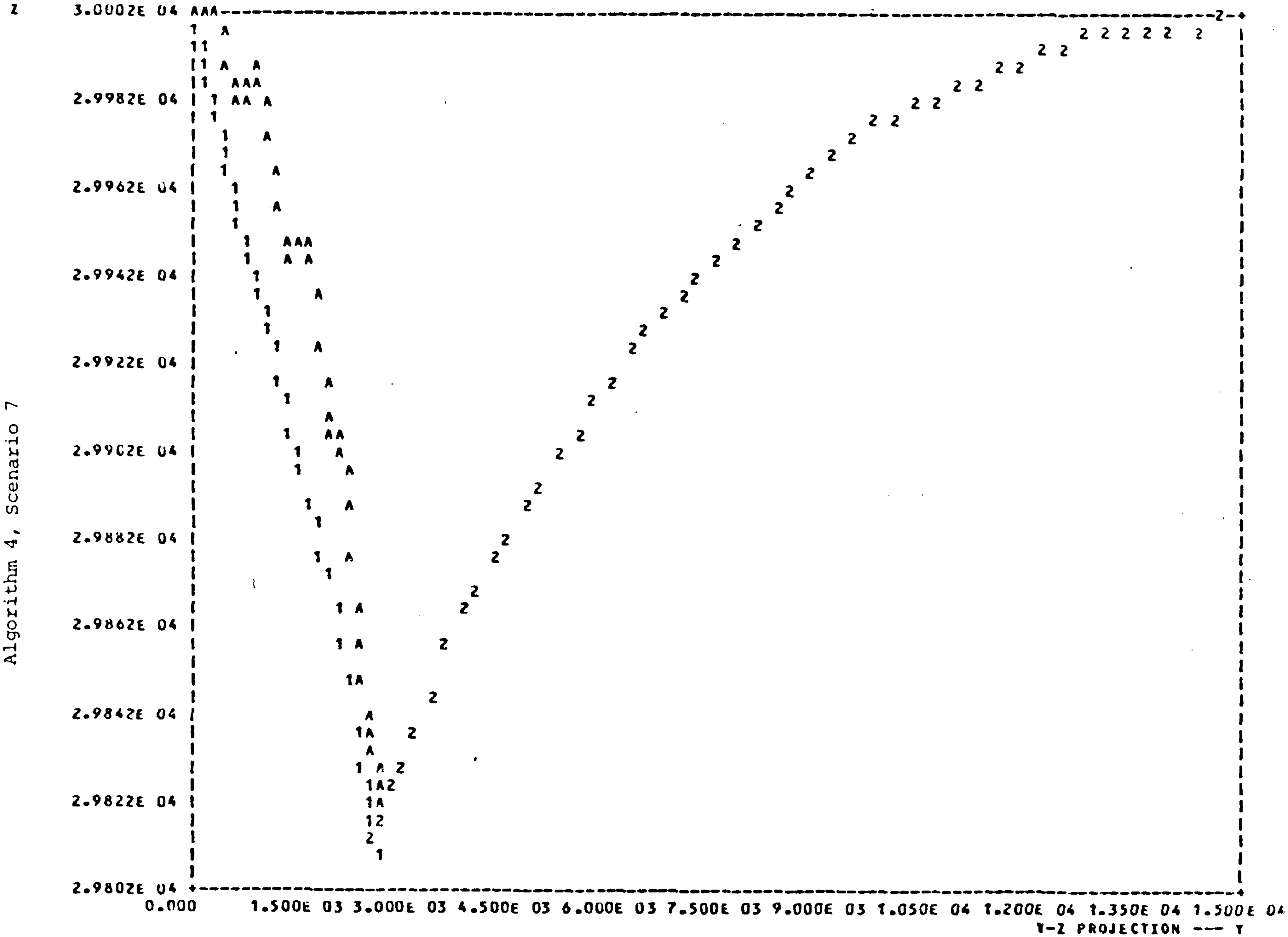


Figure 4-11d
Algorithm 4, Scenario 7

5.0 A Game Theoretic Model for Determining Aircraft Evasion Strategies Against a Multiple Missile Threat

Introduction

In this Chapter we consider a missile evasion problem formulated in terms of one aircraft (the evader) and two guided missiles. Based on various assumptions pertaining to dynamics, information patterns, and optimization criteria the existence, structure and behavior of a set of optimal evasion strategies are delineated. This study encompasses problem formulations with linear and nonlinear dynamics. In each case, the evader is assumed to know the guidance law for each pursuing missile. In this study two general classes of optimization criteria are considered: (i) fixed terminal time criteria and (ii) free terminal time criteria. In the former case, the evader seeks to maximize the terminal miss distance between himself and each pursuer. In the latter case, the evader seeks to maximize the distance of closest approach between himself and each pursuer. This decision problem is most naturally formulated as a multi-criterion or vector valued optimization problem. A game theoretic approach is taken in solving this class of problems.

The results in this chapter represent part of a Ph.D. Dissertation in Systems Engineering (Huling, 1979) and are the basis for a paper (Huling & Mintz, 1977). In the numbering convention for the following subsections in this chapter, we have omitted the leading (5.) designation.

1. OPTIMAL EVASION STRATEGIES AGAINST MULTIPLE MISSILES: PART I - FOR CRITERIA WITH A FIXED TERMINAL TIME

1.1 Introduction:

We consider a missile evasion problem formulated in terms of one aircraft (the evader) and two guided missiles. We delineate the existence, structure, and behavior of optimal evasion strategies for this problem based on the following assumptions pertaining to dynamics, information patterns, and optimization criteria.

Dynamics:

(a) The evader's and pursuer's dynamics are each linear, i.e.,

$$\dot{x}_e = F_e x_e + G_e u_e$$

$$\dot{x}_{pi} = F_{pi} x_{pi} + G_{pi} u_i, \quad i = 1,2.$$

(b) The evader's and pursuer's dynamics are each nonlinear, i.e.,

$$\dot{x}_e = f_e(x_e, u_e, t)$$

$$\dot{x}_{pi} = f_{pi}(x_{pi}, u_i, t), \quad i = 1,2.$$

Information:

Each pursuer ($i = 1,2$) uses a given feedback control law for its guidance strategy. The evader's a priori information includes a complete description of each dynamical system including initial state information.

Optimization Criterion:

The evader seeks to maximize the "miss distance" between himself and each pursuer at a given terminal time T. Since there are two pursuers, the evader is faced with a multi-criterion or vector valued optimization problem. One approach to solving this multi-criterion problem is to seek an evasion strategy

which maximizes the minimum terminal miss distance. This maxmin or game theoretic approach requires that we solve a saddle-point problem to obtain the optimal evasion strategy.

In order to obtain a physically meaningful solution, we must either modify the original problem statement to include a set of constraints on the evader's permissible controls, or modify the pay-off functions to include a cost to the evader for using energy for evasive maneuvering. We shall take this later approach initially and assume that the evader's control function is weighted quadratically in each of the individual criteria. In section 3 and subsequent sections, we consider optimal evasion problems with a variety of constraints on the evader's permissible controls.

2. AN OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND QUADRATIC COST ON CONTROL

2.1 Mathematical Problem Statement:

Let $x \triangleq (x_e, x_{p_1}, x_{p_2})'$,

then one obtains

$$\dot{x} = Fx + G_0u_0 + G_1u_1 + G_2u_2, \quad x(0) \triangleq x_0$$

where F and G_i $i = 0,1,2$ denote partitioned matrices defined in terms of F_e, F_{pi}, G_e, and G_i $i = 1,2$, and where $u_0 = u_e$.

We have assumed

$$u_i = S_i x; \quad i = 1,2.$$

Hence

$$\dot{x} = \tilde{F}x + G_0u_0; \quad x(0) = x_0$$

where $\tilde{F} = F + G_1S_1 + G_2S_2$.

Define $J_1(u_0,x_0)$ and $J_2(u_0,x_0)$ by:

$$J_1 = x'(T)C_1x(T) - \int_0^T u_0'D_0u_0\,dt$$

$$J_2 = x'(T)C_2x(T) - \int_0^T u_0'D_0u_0\,dt$$

How can J_i be

where $C_i \geq 0$ and $D_0 > 0$.

In what follows, we shall consider x_0 a fixed vector and write $J_i(u_0,x_0)$ as

$J(u_0,i)$ $i = 1,2$.

2.2 Game Theoretic Formulation:

Let the space of permissible open loop controls u_0 be $L^2[0,T]$ and let $I \triangleq \{1,2\}$. Let

$$J: L^2[0,T] \times I \to R$$

denote the kernel of an abstract game.

Consider the following saddle-point problem (SPP):

SPP1: Does $\max\limits_{u_0 \in L^2[0,T]} \min\limits_{i \in I} J(u_0,i) = \min\limits_{i \in I} \max\limits_{u_0 \in L^2[0,T]} J(u_0,i)$?

Discussion (SPP1):

Due to the discrete nature of the set $I = \{1,2\}$, SPP1 will generally not have a saddle-point in pure strategies. This situation can often be resolved however by considering mixed strategies over I. This leads naturally to SPP2.

SPP2:

Let $p \in [0,1]$ and define $\tilde{J}(u_0,p) \triangleq pJ(u_0,1) + (1-p)J(u_0,2)$.

Does

$$\max_{u_0 \in L^2[0,T]} \min_{p \in [0,1]} \tilde{J}[u_0,p] = \min_{p \in [0,1]} \max_{u_0 \in L^2[0,T]} \tilde{J}[u_0,P]?$$

Discussion (SPP2):

For conceptual reasons, it is useful to consider p as the probability that strategy $i = 1$ is played. Then ?

The missile [illegible]
or [illegible]

$$E[J(u_0,i)] = \tilde{J}(u_0,p),\quad \tilde{J}(u_0,p) = pJ_1 + (1-p)J_2$$

$$= x'(T)\tilde{C}x(T) - \int_0^T u_0' D_0 u_0 dt,$$

where $\tilde{C} = pC_1 + (1-p)C_2$.

In order to investigate the properties of the game denoted by $\{\tilde{J}(u_0,p),\ L^2[0,T],\ [0,1]\}$ it is useful to rewrite $\tilde{J}(u_0,p)$ as follows. First observe that

$$x(T) = Lx_0 + \mathcal{L}u_0$$

where $L \triangleq \Phi(T,0)$ and Φ is the state transition matrix associated with $\tilde{F}$, and where $\mathcal{L}:L^2[0,T]\to R^n$ is the linear operator defined by

$$\mathcal{L}u_0 = \int_0^T \Phi(T,\tau)G_0 u_0 d\tau.$$

Hence we can write that

$$\tilde{J}(u_0,p) \triangleq \|Lx_0 + \mathcal{L}u_0\|^2_{\tilde{C}} - \|u_0\|^2_{D_0},$$

where the first norm is w.r.t. R^n and the second norm is w.r.t. $L^2[0,T]$.

2.3 <u>Properties of $\tilde{J}:L^2[0,T]x[0,1]\to R$</u>:

<u>Observation 1</u>:

If for all $p\varepsilon[0,1]$, the matrix Riccati equation

$$\dot{R} = -\tilde{F}'R - R\tilde{F} - RG_0 D_0^{-1} G_0' R;\quad R(T) = \tilde{C}$$

has a <u>bounded</u> solution on $[0,T]$, then $\tilde{J}[u_0,p]$ is concave in u_0 for every $p\varepsilon[0,1]$.

We will denote the existence of bounded solutions to this Riccati equation for all $p\varepsilon[0,1]$ as <u>Condition 1</u>.

An Aside: If Condition 1 does not hold, a relaxed version may hold on a subset of [0,1], i.e., for only some values of p.

Observation 2:

If Condition 1 holds, then for fixed p, $\tilde{J}[u_0,p]$ is weakly upper semi-continuous in u_0 on $L^2[0,T]$.

Observation 3:

For fixed $u_0 \varepsilon L^2[0,T]$, $\tilde{J}[u_0,p]$ is continuous in p on [0,1].

Observation 4:

For fixed $u_0 \varepsilon L^2[0,T]$, $\tilde{J}(u_0,p)$ is convex (affine) in p on [0,1].

Observation 5:

[0,1] is a compact convex subset of R.

Observation 6:

$L^2[0,T]$ is convex.

Lemma 1:

If Condition 1 holds, then $\sup_{u_0} \inf_{p} \tilde{J}(u_0,p) = \inf_{p} \sup_{u_0} \tilde{J}(u_0,p)$.

Proof: It follows from Observations 1-6 that $\tilde{J}(u_0,p)$ is quasi-concave-quasi-convex in (u_0,p) and u.s.c. - l.s.c. Further, $L^2[0,T]$ and [0,1] are convex spaces, and [0,1] is compact in R. Therefore, Lemma 1 follows as a consequence of Sion's Theorem [Sion: 1958].

Lemma 2:

If Condition 1 holds, then $\min_{p} \sup_{u_0} \tilde{J}(u_0,p) = \sup_{u_0} \min_{p} \tilde{J}(u_0,p)$

i.e., there exists a value of p (say p*) such that

$$\sup_{u_0} \tilde{J}(u_0,p^*) = \inf_{p} \sup_{u_0} \tilde{J}(u_0,p).$$

Proof: Lemma 2 is a consequence of Lemma 1 and the compactness of [0,1].

Observation 7:

If Condition 1 holds, there exists a value of u_0 (say u_0*) such that

$$\sup_{u_0} \tilde{J}(u_0,p^*) = \tilde{J}(u_0^*,p^*).$$

An Aside: Observation 7 is a consequence of the assumption of Condition 1.

Theorem 1: The pair (u_0^*,p^*) defined through Lemma 2 and Observation 7 constitute a saddle-point for $\tilde{J}(u_0,p)$.

Proof: The verification that $\tilde{J}(u_0,p^*) \leq \tilde{J}(u_0^*,p^*) \leq \tilde{J}(u_0^*,p)$ follows directly from the definition of (u_0^*,p^*) in Lemma 2 and Observation 7.

3. AN OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND A TOTAL ENERGY CONSTRAINT:

3.1 Problem Statement:

Let

$$\dot{x} = \tilde{F}x + G_0 u_0 \;;\; x(0) \triangleq x_0 \dagger$$

where

$$u_0 \varepsilon \Omega_e;$$

$$\Omega_e = \{u_0 \varepsilon L^2[C,T] : \|u_0\|^2_{D_e} \leq e^2\}.$$

Define J_1 and J_2 by

$$J_1 = x'(T)C_1 x(T) - \int_0^T u_0' D_0 u_0 dt$$

$$J_2 = x'(T)C_2 x(T) - \int_0^T u_0' D_0 u_0 dt.$$

Consider the following Maxmin Problem:

$$\max_{u_0 \varepsilon \Omega_e} \; \min_{i \varepsilon I} \; J(u_0, i) \;.$$

The sole difference between this problem and that considered in the previous section is the restriction of u_0 to the weakly compact set $\Omega_e \subset L^2[0,T]$,

$$\Omega_e \triangleq \{u_0 \varepsilon L^2[0,T] : \|u_0\|^2_{D_e} \leq e^2\}.$$

led ?

Reasoning as before, we are lead to consider

$$\max_{u_0 \varepsilon \Omega_e} \; \min_{p \varepsilon [0,1]} \; \tilde{J}(u_0, p)$$

$\dagger$ $\tilde{F}$ was defined in Section 2.

where $\tilde{J}(u_0,p) = pJ(u_0,1) + (1-p)\ J(u_0,2)$.

3.2 Problem Solution:

Consider the following saddle-point problem (SPP3):

Does

$$\max_{u_0 \varepsilon \Omega_e} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p) = \min_{p \varepsilon [0,1]} \max_{u_0 \varepsilon \Omega_e} \tilde{J}(u_0,p)?$$

Observation 1:

Ω_e is a convex weakly compact subset of $L^2[0,T]$.

Condition 1:

The Riccati matrix differential equation

$$\dot{R} = -\tilde{F}'R - R\tilde{F} - RG_0D_0^{-1}G_0R$$

$$R(T) = \tilde{C} \triangleq pC_1 + (1-p)C_2$$

has a bounded solution on $[0,T]$ for all $p\varepsilon[0,1]$.

Now using the results associated with SPP2, we have:

Theorem 1:

If Condition 1 holds, then

$$\max_{u_0 \varepsilon \Omega_e} \min_{p \varepsilon [0,1]} \tilde{J}[u_0,p] = \min_{p \varepsilon [0,1]} \max_{u_0 \varepsilon \Omega_e} \tilde{J}[u_0,p].$$

Proof:

The proof again follows from Sion's Theorem [Sion: 1958] by noting that:

(a) Ω_e is a convex weakly compact subset of $L^2[0,T]$ and $[0,1]$ is a convex compact subset of R.

(b) $\tilde{J}:\Omega_e \times [0,1] \to R$

is quasi-concave-convex, weakly u.s.c. in u_0 for fixed p, and continuous in p for fixed u_0.

3.3 Solution Structure:

Lemma 1:

Let (u_0^*, p^*) denote any saddle-point solution to SPP3, then

$$u_0^* = \arg\max_{u_0 \in \Omega_e} \tilde{J}(u_0, p^*)$$

where:

$$u_0^* = (D_0 + \alpha^* D_e)^{-1} G_0' kx,$$

$$\dot{k} = -\tilde{F}'k - k\tilde{F} - kG_0(D_0 + \alpha^* D_e)^{-1} G_0' k,$$

$$k(T) = \tilde{C} = p^* C_1 + (1-p^*) C_2 ,$$

and $\alpha^* \left(\int_0^T u_0'^* D_e u_0^* dt - e^2 \right) = 0$,

$\alpha^* \geq 0$.

Proof: Lemma 1 follows as a consequence of the Linear-Quadratic (LQ) nature of the problem setting and the application of standard multiplier theory. (See for example [Luenberger: 1969, P. 217].)

4. A SECOND OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND A TOTAL ENERGY CONSTRAINT:

4.1 Problem Statement:

Let

$$\dot{x} = \tilde{F}x + G_0 u_0; \qquad x(0) \triangleq x_0$$

where

$$u_0 \varepsilon \Omega_e;$$

$$\Omega_e = \{u_0 \varepsilon L^2[0,T]: \|u_0\|^2_{D_e} \leq e^2\}.$$

Define J_1 and J_2 by

$$J_1 = x'(T)C_1 x(T)$$

$$J_2 = x'(T)C_2 x(T).$$

Consider the following Maxmin Problem:

$$\max_{u_0 \varepsilon \Omega_e} \quad \min_{i \varepsilon I} \quad J(u_0, i).$$

The sole difference between this problem and that considered in the previous section is the elimination of the quadratic terms in u_0 from the pay-off functions J_1 and J_2.

Reasoning as before, we are lead to consider

$$\max_{u_0 \varepsilon \Omega_e} \quad \min_{p \varepsilon [0,1]} \quad \tilde{J}(u_0, p)$$

where

$$\tilde{J}(u_0,p) = pJ(u_0,1) + (1-p)J(u_0,2)$$

$$= x'(T)[pC_1 + (1-p)C_2]x(T).$$

4.2 Problem Solution:

Consider the following saddle-point problem (SPP4):

Does

$$\max_{u_0 \varepsilon \Omega_e} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p) = \min_{p \varepsilon [0,1]} \max_{u_0 \varepsilon \Omega_e} \tilde{J}(u_0,p)?$$

Lemma 1: For fixed $p \varepsilon [0,1]$, $J(u_0,p)$ is w-continuous on Ω_e.

Proof: For notational simplicity, we consider the scalar case, i.e., $\dim(x) = \dim(u_0) = 1$.

We note that

$$x(T) = \Phi(T,0)x_0 + \int_0^T \Phi(T,\tau)G_0 u_0 d\tau$$

where Φ is the state transition function associated with $\tilde{F}$. Hence, $x(T)$ can be expressed as

$$x(T) = a + \langle b,u_0 \rangle$$

where $a \varepsilon R$, $b \varepsilon L^2[0,T]$ and $\langle \cdot , \cdot \rangle$ denotes the standard inner product on $L^2[0,T]$.

(Here we are of course assuming that

$\Phi(T,t)G_0 \; \varepsilon \; L^2[0,T]$).

Therefore, $x^2(T)$ is w-continuous for all $u_0 \varepsilon L^2[0,T]$ since $x(T)$ is w-continuous in u_0. Finally,

$$\tilde{J}(u_0,p) = x^2(T)[pC_1 + (1-p)C_2],$$

and therefore, $\tilde{J}(u_0,p)$ is w-continuous on $L^2[0,T]$ for fixed $p \varepsilon [0,1]$.

Observation 1:

$\tilde{J}(u_0,p) \triangleq x^2(T)[pC_1 + (1-p)C_2]$ is continuous in the pair (u_0,p) when $\tilde{J}:\Omega_e \times [0,1] \rightarrow R$, where Ω_e is endowed with the weak topology. This follows from the product structure of $\tilde{J}$, where we observe that the product of the limits of two sequences is the limit of product of the individual sequences.

Observation 2:

$\tilde{J}:\Omega_e \times [0,1] \rightarrow R$ is continuous in the pair (u_0,p) in the general case - $\dim(x) = n$, $\dim(u) = r$; where Ω_e is endowed with the weak topology. Observation 2 follows by noting that in the general case the components of $x(T)$ can be considered as a linear combination of underlying linear functionals.

Theorem 1:

Let M_u and M_p denote respectively the set of all probability measures μ_e (respectively μ_p) on Ω_e (respectively [0,1]). Then,

$$\max_{\mu_e \varepsilon M_u} \ \min_{\mu_p \varepsilon M_p} E[\tilde{J}(u_0,p)] = \min_{\mu_p \varepsilon M_p} \ \max_{\mu_e \varepsilon M_u} E[\tilde{J}(u_0,p)],$$

where the expectation operation is with respect to the measures μ_e and μ_p.

Comment:

The interpretation of this theorem is that the saddle-point problem, (SPP4) has a solution in mixed strategies. In what follows, we will in fact show that (SPP4) has a solution in pure strategies. *To do this, we will need* Theorem 1 as a preliminary result.

Proof of Theorem 1:

Since $\tilde{J}(u_0,p)$ is continuous in the pair (u_0,p) and Ω_e and $[0,1]$ are respectively w-compact and compact then Theorem 1 is a direct consequence of a general saddle-point theorem [See [Owen: 1968] (Theorem IV.6.1)].

Theorem 2:

Let (μ_e^*,μ_p^*) denote a pair of optimal mixed strategies with respect to the game defined by $\tilde{J}:\Omega_e x[0,1]\to R$. Let $p^* \triangleq E_{\mu^*_p}[p]$. If $\arg\max_{u_0\epsilon\Omega_e} \tilde{J}(u_0,p^*)$ is essentially unique, then, the pair (μ^*_e,μ^*_p) are one point, i.e., μ^*_e and μ^*_p correspond to pure strategies.

Proof:

Since $\tilde{J}[u_0,p]$ is affine in p for fixed u_0, $p^* \triangleq E_{\mu^*_p}[p]$ can be viewed as an optimal pure strategy since

$$E_{\mu^*_p}[\tilde{J}(u_0,p)] = \tilde{J}(u_0,p^*).$$

Let $u^*_0 = \arg\max_{u_0\epsilon\Omega_e} \tilde{J}(u_0,p^*)$. Since u^*_0 is essentially unique by assumption, the pure strategy u^*_0 is equivalent to the optimal mixed strategy μ^*_e.

Lemma 1:

$$u_0^* = \frac{1}{\alpha^*} D_e^{-1} G_0' \tilde{R} x$$

where

$$\dot{\tilde{R}} = - \tilde{F}'\tilde{R} - \tilde{R}\tilde{F} - \frac{1}{\alpha^*} \tilde{R} G_0 D_e^{-1} G_0' \tilde{R}$$

$$\tilde{R}(T) \triangleq \tilde{C} = p^* C_1 + (1-p^*) C_2, \text{ and}$$

$$\int_0^T u_0^* D_e u_c^* dt = e^2.$$

Proof: Lemma 1 follows as a consequence of the Linear-Quadratic (LQ) nature of the problem setting and the application of standard multiplier theory. (See for example [Luenberger: 1969, P. 220].)

5. AN OPTIMAL EVASION PROBLEM WITH LINEAR DYNAMICS AND HARD CONSTRAINTS ON CONTROL:

5.1 Problem Statement:

Let

$$\dot{x} = Fx + G_0 u_0 ; \quad x(0) \triangleq x_0$$

where

$u_0 \in \Omega_b$; $\Omega_b \triangleq \{ u_0 \varepsilon L^2[0,T]: \ a_i \le u_i(t) \le b_i \}$, and u_i denotes the i^{th} component of the vector u_0.

Define J_1 and J_2 by

$$J_1 = x'(T) C_1 x(T)$$

$$J_2 = x'(T) C_2 x(T) \quad .$$

Consider the following Maxmin Problem:

$$\max_{u_0 \varepsilon \Omega_b} \ \min_{i \varepsilon I} \ J(u_0, i).$$

The sole difference between this problem and that considered in the previous section is the replacement of Ω_e with Ω_b.

Reasoning as before, we are lead to consider

$$\max_{u_0 \varepsilon \Omega_b} \ \min_{p \varepsilon [0,1]} \ \tilde{J}(u_0, p)$$

where $\tilde{J}(u_0,p) = x'(T)[pC_1 + (1-p)C_2]x(T)$.

5.2 Problem Solution:

Consider the following saddle-point problem (SPP5):

Does

$$\max_{u_0 \varepsilon \Omega_b} \ \min_{p \varepsilon [0,1]} \ \tilde{J}(u_0, p) = \min_{p \varepsilon [0,1]} \ \max_{u_0 \varepsilon \Omega_b} \ \tilde{J}(u_0, p)?$$

In order to answer SPP5, we will first formulate and solve a related saddle-point problem SPP6:

We begin with the following definition:

Definition:

$$\underline{\overline{X}}(T,x_0) \triangleq \{x(T)\varepsilon E^n: \dot{x} = \tilde{F}x + G_0 u_0;\ x(0) \triangleq x_0;\ u_0 \varepsilon \Omega_b\}$$

Observation 1:

$\underline{\overline{X}}(T,x_0)$ is a compact convex subset of E^n.

Consider the following saddle-point problem (SPP6):

Does

$$\max_{x(T)\varepsilon\underline{\overline{X}}(T,x_0)} \ \min_{p\varepsilon[0,1]} \ x'(T)\tilde{C}x(T) = \min_{p\varepsilon[0,1]} \ \max_{x(T)\varepsilon\underline{\overline{X}}(T,x_0)} \ x'(T)\tilde{C}x(T) \ ?$$

Theorem 1:

Let M_x and M_p denote respectively the set of all probability measures μ_x (respectively μ_p) on $\underline{\overline{X}}(T,x_0)$ (respectively [0,1]). Then,

$$\max_{\mu_x\varepsilon M_x} \ \min_{\mu_p\varepsilon M_p} \ E[x'(T)\tilde{C}x(T)] = \min_{\mu_p\varepsilon M_p} \ \max_{\mu_x\varepsilon M_x} \ E[x'(T)\tilde{C}x(T)],$$

where the expectation operation is with respect to the measures μ_x and μ_p.

Proof:

The proof follows by noting that $\underline{\overline{X}}(T,x_0)$ is compact in E^n, [0,1] is compact in R, and $x'(T)\tilde{C}x(T)$ is continuous in the pair (x(T),p). Hence the conclusion follows as a consequence of Theorem IV.6.1 [Owen: 1968].

Theorem 2:

Let (μ_x^*,μ_p^*) denote a pair of optimal mixed strategies with respect to the game defined by:

$$k: \underline{\overline{X}}(T,x_0) \times [0,1] \to R,\ k(x(T),p) = x'(T)\tilde{C}x(T).$$

Let $p^* \triangleq E_{\mu_p^*}[p]$. If $x^*(T) = \underset{\underline{\overline{X}}(T,x_0)}{\arg\max}\ k(x(T),p^*)$ is unique, then the pair (μ_x^*,μ_p^*) are one-point, i.e., μ_x^* and μ_p^* correspond to pure strategies.

Proof:

Since $k(x(T),p)$ is affine in p for fixed $x(T)$, $p^* \triangleq E_{\mu_p^*}[p]$ can be viewed as an optimal pure strategy since

$$E_{\mu_p^*}[k(x(T),p)] = k(x(T),p^*).$$

$$\text{Let } x^*(T) \triangleq \underset{x(T)\epsilon\underline{\overline{X}}(T,x_0)}{\arg\max}\ k(x(T),p^*).$$

Since $x^*(T)$ is unique by assumption, the pure strategy $x^*(T)$ is equivalent to the optimal mixed strategy μ_x^*.

Theorem 3:

Let u_0^* denote any control in Ω_b which achieves $x(T) = x^*(T)$, then

$$\max_{u_0\epsilon\Omega_b}\ \min_{p\epsilon[0,1]}\ \tilde{J}(u_0,p) = \min_{p\epsilon[0,1]}\ \max_{u_0\epsilon\Omega_b}\ \tilde{J}(u_0,p) = \tilde{J}(u_0^*,p^*),$$

i.e. (u_0^*,p^*) is a saddle-point pair for SPP5. [$x^*(T)$ and p^* are defined in Theorem 2].

Proof: The proof follows directly by noting that $(x^*(T),p^*)$ is a saddle-point pair for the game defined by:

$$k:\underline{\overline{X}}(T,x_0) \times [0,1] \to R,$$

$$k(x(T),p) = x'(T)\tilde{C}x(T).$$

6. AN OPTIMAL EVASION PROBLEM WITH NONLINEAR DYNAMICS AND HARD CONSTRAINTS ON CONTROL:

6.1 Problem Statement:

Let $\dot{x} = f(x,u_0,t)$ $x(0) \triangleq x_0$ where $u_0 \varepsilon \Omega$;

$\Omega = \{u_0 : u_0(t) \varepsilon E^r;\ u_i$ Lebesgue measurable on $[0,T]$, $a_i \leq u_i(t) \leq b_i$, $i = 1,\ldots,r\}$.

Define J_1 and J_2 by

$$J_1 = x'(T)C_1 x(T)$$

$$J_2 = x'(T)C_2 x(T) .$$

Consider the following maxmin problem:

$$\max_{u_0 \varepsilon \Omega} \min_{i \varepsilon I} J(u_0,i).$$

The sole difference between this problem and that considered in the previous section is that the dynamics are now allowed to be possibly nonlinear.

Reasoning as before we are lead to consider

$$\max_{u_0 \varepsilon \Omega} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p)$$

where $\tilde{J}(u_0,p) = x'(T)\ [pC_1 + (1-p)C_2]x(T)$.

Consider the following saddle-point problem (SPP7):

Does

$$\max_{u_0 \varepsilon \Omega} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p) = \min_{p \varepsilon [0,1]} \max_{u_0 \varepsilon \Omega} \tilde{J}(u_0,p)?$$

In order to answer SPP7, we will first formulate and solve a related saddle-point problem SPP8. This approach parallels our analysis in section 5.

We begin with the following definition:

$$\underline{\overline{X}}(T,x_0) \triangleq \{x(T)\varepsilon E^n: \dot{x} = f(x, u_0,t);\ x(0) = x_0;\ u_0\varepsilon\Omega\}$$

Assumption 1:

Assume $\underline{\overline{X}}(T,x_0)$ is compact in E^n.

Consider the following saddle-point problem (SPP8):

Does

$$\max_{u_0\varepsilon\Omega}\ \min_{p\varepsilon[0,1]}\ \tilde{J}(u_0,p) = \min_{p\varepsilon[0,1]}\ \max_{u_0\varepsilon\Omega}\ \tilde{J}(u_0,p)?$$

Theorem 1:

Let M_x and M_p denote respectively the set of all probability measures μ_x (respectively μ_p) on $\underline{\overline{X}}(T,x_0)$ (respectively [0,1]). Then,

$$\max_{\mu_x\varepsilon M_x}\ \min_{\mu_p\varepsilon M_p}\ E[x'(T)\tilde{C}x(T)] = \min_{\mu_p\varepsilon M_p}\ \max_{\mu_x\varepsilon M_x}\ E[x'(T)\tilde{C}x(T)]$$

where the expectation operation is with respect to the measures μ_x and μ_p.

Proof:

The proof of this result is precisely that of Theorem 1, section 5.

Theorem 2:

Let (μ_x^*,μ_p^*) denote a pair of optimal mixed strategies with respect to the game defined by:

$k:\underline{\overline{X}}(T,x_0) \times [0,1] \to R,$

$k(x(T),p) = x'(T)\tilde{C}x(T).$

Let $p^* = E_{\mu_p^*}[p]$. If $x^*(T) \triangleq \arg\max_{\underline{X}(T,x_0)} k(x(T),p^*)$ is unique then the pair (μ_x^*,μ_p^*) are one-point i.e., μ_x^* and μ_p^* correspond to pure strategies.

Proof:

The proof of this result is precisely that of Theorem 2, section 5.

Theorem 3 :

Let u_0^* denote any control in Ω which achieves $x(T) = x^*(T)$, then

$$\max_{u_0 \in \Omega} \min_{p\in[0,1]} \tilde{J}(u_0,p) = \tilde{J}(u_0^*,p^*) = \min_{p\in[0,1]} \max_{u_0\in\Omega} \tilde{J}(u_0,p).$$

Hence, the pair (u_0^*,p^*) is a saddle-point for the game defined by $\tilde{J}(u_0,p)$ (SPP7), where $x^*(T)$ and p^* are defined in Theorem 3.

Proof:

The proof of this result is precisely that of Theorem 3, section 5.

6.3 Some Comments on Theorem 1-3, Sections 5 & 6:

In order to extend the saddle-point results for the case of linear dynamics (section 5) to include nonlinear dynamics (section 6), we have assumed that the set of attainment $\underline{\overline{X}}(T,x_0)$ is compact in E^n (Assumption 1, section 6). In this section we will consider sufficient conditions to guarantee the compactness of $\underline{\overline{X}}(T,x_0)$. The following results are due to Filippov (Filippov; 1962).

We begin with several definitions:

Let $\dot{x} = f(x,u,t)$, $x(0) = x_0$ where $\dim(x) = n$, $\dim(u) = r$. Suppose each component u_i of u is a measurable function of t satisfying $a_i \leq u_i(t) \leq b_i$, $0 \leq t \leq T$. Let U denote the permissible range of u, i.e. $U = \prod_{i=1}^{r} [a_i, b_i]$. Suppose that f is continuous in all arguments, and is continuously differentiable with respect to x, and that

$$x'f(x,u,t) \leq C[1 + \|x\|^2] \text{ for some } C \text{ and all } t, x, u \in U.$$

Let $R(t,x)$ denote the following set valued function:

$$R(t,x) = \{f(x,u,t): \quad u \in U\}$$

<u>Theorem 4 (Filippov)</u>:

If f, and U are defined as above, and $R(t,x)$ is convex for all t, $0 \leq t \leq T$, and $x \in E^n$, then $\underline{X}(T,x_0)$ is compact.

Comment: Theorem 4 is also true when $u(t) \in U(t)$, where $U(t)$ is compact in E^r, and the set valued function $U(\cdot)$ is t-continuous in the Hausdorff Topology, for all t, $0 \leq t \leq T$.

Proof: See [Filippov; 1962], or [Hermes and LaSalle; 1969].

7. AN OPTIMAL EVASION PROBLEM WITH NONLINEAR DYNAMICS, HARD CONSTRAINTS ON CONTROL, AND A GENERAL TERMINAL COST FUNCTION:

7.1 Problem Statement:

Let $\dot{x} = f(x,u_0,t)$, $x(0) = x_0$, where $u_0 \varepsilon \Omega$;

$\Omega \triangleq \{u_0 : u_0(t) \varepsilon E^r;\ u_i$ Lebesgue measurable on $[0,T]$, $a_i \leq u_i(t) \leq b_i$, $i = 1,2, \ldots, r\}$, and $U \triangleq \prod_{i=1}^{r} [a_i, b_i] \subset E^r$.

Let $J_1(x(T))$ and $J_2(x(T))$ denote terminal cost functions.

Assumptions:

A_1 : J_1 and J_2 are continuous maps of E^n into R.

A_2 : f: $E^n \times U \times [0,T] \to E^n$ is continuous in all arguments, is continuously differentiable in its first argument (x);

$x' f(x,u,t) \leq C[1 + \|x\|^2]$ for some C and all $x \varepsilon E^n$, $u \varepsilon U$, and $t \varepsilon [0,T]$.

A_3 : If $R(t,x) \triangleq \{f(x,u,t) : u \varepsilon U\}$, then the set $R(t,x)$ is convex for all pairs (t,x) where $0 \leq t \leq T$, $x \varepsilon E^n$.

A_4 : Let $\underline{\overline{X}}(t,x_0)$ and $\underline{\overline{Y}}$ denote

$$\underline{\overline{X}}(T,x_0) \triangleq \{x(T) \varepsilon E^n : \dot{x} = f(x,u_0,t),\ x(0) = x_0,\ u_0 \varepsilon \Omega\}$$

$$\underline{\overline{Y}} \triangleq \{(J_1(x(T)),\ J_2(x(T))) \varepsilon E^2 :\ x(T) \varepsilon \underline{\overline{X}}(T,x_0)\},$$

then $\underline{\overline{Y}}$ is a convex set.

Consider the following maxmin problem:

$$\max_{u_0 \varepsilon \Omega} \ \min_{i \varepsilon I} \ J_i(x(T)).$$

The sole difference between this problem and that considered in the previous section is that the functions $J_i(x(T))$ are now allowed to be arbitrary continuous mappings.

Reasoning as before, we are lead to consider

$$\max_{u_0 \varepsilon \Omega} \min_{p \varepsilon [0,1]} \tilde{J}(u_0, p)$$

where $\tilde{J}(u_0,p) = pJ_1(x(T)) + (1-p)J_2(x(T))$.

7.2 Problem Solution:

Consider the following saddle-point problem (SPP9):

Does

$$\max_{u_0 \varepsilon \Omega} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p) = \min_{p \varepsilon [0,1]} \max_{u_0 \varepsilon \Omega} \tilde{J}(u_0,p)?$$

Theorem 1:

Under Assumptions $A_1 - A_4$, SPP9 has a solution in pure strategies.

Proof: We begin with several observations.

Observation 1:

The set $\bar{Y}$ is compact in E^2. This observation follows by noting that $\bar{Y}$ is the image of $\bar{X}(T,x_0)$ under $\hat{J}$, where $\hat{J}(x(T)) = (J_1(x(T)), J_2(x(T)))$, $\hat{J}$ is continuous (Assumption A_1), and $\bar{X}(T,x_0)$ is compact in E^n (Theorem 4, section 6).

Observation 2:

Let $k: \bar{Y} \times [0,1] \to R$, $k(y,p) = py_1 + (1-p)y_2$.

Since k is continuous in (y,p), convex (affine) in p for fixed y, concave (affine) in y for fixed p, and $\underline{\overline{Y}}$, [0,1] are compact convex sets then by Sion's Theorem [Sion: 1958], the game defined by $\{k,\underline{\overline{Y}},[0,1]\}$ has a solution in pure strategies.

Hence, the proof of Theorem 1 is complete by noting that the games $\{\tilde{J},\Omega,[0,1]\}$ and $\{k,\underline{\overline{Y}},[0,1]\}$ are equivalent. Therefore, if (y^*,p^*) denotes any solution to $\{k,\underline{\overline{Y}},[0,1]\}$, (u^*_0,p^*) denotes a saddle-point solution for $\{\tilde{J},\Omega,[0,1]\}$ when u^*_0 is any control which achieves $y^* = \hat{J}(x^*(T))$.

<u>Comment</u>:

Assumption A_4 of this section in a sense replaces the assumption of the uniqueness of $x^*(T) = \arg\max_{\underline{X}(T,x_0)} k(x(T),p^*)$ which appears in Theorem 2 of sections 5 and 6.

8. OPTIMAL EVASION STRATEGIES AGAINST MULTIPLE MISSILES: PART II - FOR CRITERIA WITH A FREE TERMINAL TIME

8.1 Introduction:

We consider a missile evasion problem formulated in terms of one aircraft (the evader) and two guided missiles. We delineate the existence, structure, and behavior of optimal evasion strategies for this problem based on the following assumptions pertaining to dynamics, information patterns, and optimization criteria.

Dynamics:

The evader's and pursuer's dynamics are each nonlinear, i.e.,

$$\dot{x}_e = f_e(x_e, u_e, t)$$

$$\dot{x}_{pi} = f_{pi}(x_{pi}, u_i, t), \quad i = 1,2.$$

Information:

Each pursuer ($i = 1,2$) uses a given feedback control law for its guidance strategy. The evader's a priori information includes a complete description of each dynamical system including initial state information.

Optimization Criterion:

The evader seeks to maximize the "distance of closest approach" between himself and each pursuer over a given time interval $[0,T]$. Since there are two pursuers, the evader is faced with a multi-criterion or vector valued optimization problem. One approach to solving this multi-criterion problem is to seek an evasion strategy, which maximizes the minimum. This maxmin or game theoretic distance of closest approach between the evader and each pursuer over a given time interval $[0,T]$.

This approach requires that we solve a saddle-point problem to obtain the optimal evasion strategy.

In order to obtain a physically meaningful solution, we shall modify the original problem statement to include a set of constraints on the evader's permissible controls. In addition, we shall assume that for this given problem setting there exists a value of T (suitably large) such that thre is <u>no</u> need for the evader to consider strategies for $t>T$. This implies that the distance of closest approach for each missile occurs at an interior point of $[0,T]$. Hence, we have chosen the free terminal time nomenclature to describe this situation. This latter assumption is easily justified in the setting where each missile is non-thrusting after $t = 0$, and drag forces are included in the problem formulation.

In the following section, it is assumed that the reader is familiar with the material in sections 1-7 of this report.

9. AN OPTIMAL EVASION PROBLEM WITH NONLINEAR DYNAMICS, HARD CONSTRAINTS ON CONTROL, AND A GENERAL COST FUNCTION:

9.1 Problem Statement:

Let $\dot{x} = f(x,u_0,t)$, $x(0) = x_0$, where $u_0 \varepsilon \Omega$; and

$\Omega \triangleq \{u_0 : u_0(t) \varepsilon E^r$; u_i Lebesgue measurable on $[0,T]$, $a_i \leq u_i(t) \leq b_i$, $i = 1,2, . . ,r\}$, and $U = \prod_{i=1}^{r} [a_i, b_i]$.

Let $J_1(x(t))$ and $J_2(x(t))$ denote general cost functions. For example $J_i(x(t))$ could denote the actual distance between missile i and the aircraft at time t. However, the function J_i could also be chosen to depend on the relative orientation of the missile and aircraft at time t.

Assumptions:

A_1 : J_1 and J_2 are continuous maps of E^n into R.

A_2 : f: $E^n \times U \times [0,T] \rightarrow E^n$ is continuous in all arguments, is contínously differentiable in its first argument (x);

$x' f(x,u,t) \leq C[1 + \|x\|^2]$ for some C, all $x \varepsilon E^n$, $u \varepsilon U$ and $t \varepsilon [0,T]$.

A_3 : If $R(t,x) \triangleq \{f(x,u_0,t): a_i \leq u_i \leq b_i, i = 1, ..,r\}$, then the set $R(t,x)$ is convex for all pairs (t,x) where $0 \leq t \leq T$, $x \varepsilon E^n$.

A_4 : Let

$B \triangleq \{(b_1(x_0,u_0), b_2(x_0,u_0)) \varepsilon E^2 : x(0) = x_0, u_0 \varepsilon \Omega\}$,

where $b_i(x_0,u_0) \triangleq \min_{0 \leq t \leq T} J_i(x(t))$, then B is a convex set.

Consider the following Maxmin Problem:

$$\max_{u_0 \varepsilon \Omega} \ \min_{i \varepsilon I} \ b_i(x_0,u_0).$$

Reasoning as before, we are lead to consider

$$\max_{u_0 \varepsilon \Omega} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p)$$

where $\tilde{J}(u_0,p) = pb_1(x_0,u_0) + (1-p)b_2(x_0,u_0)$.

9.2 Problem Solution:

Consider the following saddle-point problem (SPP10):

Does

$$\max_{u_0 \varepsilon \Omega} \min_{p \varepsilon [0,1]} \tilde{J}(u_0,p) = \min_{p \varepsilon [0,1]} \max_{u_0 \varepsilon \Omega} \tilde{J}(u_0,p)?$$

Theorem 1:

Under Assumptions $A_1 - A_4$, SPP10 has a solution in pure strategies.

Proof: We begin with several observations.

Observation 1:

The set B is compact in E^2.

Observation 2:

Let $k: B \times [0,1] \to R$, $k(b,p) = pb_1 + (1-p)b_2$.

Since k is continuous in (b,p), convex (affine) in p for fixed b, concave (affine) in b for fixed p, and B, [0,1] are compact convex sets then by Sion's Theorem [Sion: 1958], the game defined by {k,B,[0,1]} has a solution in pure strategies.

Hence, the proof of Theorem 1 is complete by noting that the games {J,Ω, [0,1]} and {k,B,[0,1]} are equivalent. Therefore, if (b*,p*) denotes any solution to {k,B,[0,1]}, (u^*_0,p^*) denotes a saddle-point solution for $\{\tilde{J},\Omega,[0,1]\}$ when u^*_0 is any control which achieves $b^* = (b_1(x_0,u^*_0),b_2(x_0,u^*_0))$.

10. HISTORICAL PRECIS:

10.1 Games of Evasion with More Than One Pursuer:

As far as we are aware, Games of Evasion against several pursuers have not appeared extensively in the literature. Isaacs considers a very simple example of such a Differential Game on p. 148 of his celebrated monograph [1965].

10.2 Games of Evasion with a Single Pursuer:

The literature on Games of Evasion against a single pursuer is relatively extensive. We cite the following sources as examples: The monographs by Isaacs [1965]; Bryson and Ho [1969]; and Friedman [1971]. The papers by Behn and Ho [1968]; Rhodes and Luenberger [1969]; Basar and Mintz [1973]; and Poulter and Anderson [1976]. The thesis by Poulter [1975].

10.3 Optimization Problems with Vector Valued Pay-off:

The literature concerning optimization problems with vector valued pay-off functions is considerable. We cite the seminal paper by Da Cunha and Polak [1967] and the contribution by Reid and Citron [1971] as examples.

10.4 Linear-Quadratic Optimization Problems with Vector Valued Pay-off:

The problem of solving an LQ optimization problem with vector valued pay-off has been considered by Medanic and Andjelic in a sequence of papers and letters [Medanic and Andjelic: 1971, 1972a, and 1972b]. (See also the critical appraisal of this work by Ho [1971].)

The results of section 2 of this present report parallels the open-loop results obtained by Medanic and Anjelic [1971, 1972b]. The basic difference (besides the methodology) is that we consider a max min problem with weighting matrices with both positive and negative eigenvalues, whereas Medanic and Andjelic

consider a min max problem with positive and nonnegative weighting matrices. This difference leads to the possible existence of a conjugate point in the present problem setting, which in turn provides us with a natural interpretation for our Condition 1 (section 2). We note that no conjugate point phenomenon can arise in [Medanic and Andjelic: 1971, 1972b].

6.0 Conclusions and Recommendations for Further Research

Four real-time heuristic algorithms for determining aircraft evasion strategies against a multiple missile threat have been described. All four heuristic algorithms are motivated by a formal game theoretic model for multiple missile evasion. Algorithms 1 and 2 are based on "myopic" saddle point calculations which apportion the projection of the instantaneous aircraft acceleration among the normals to the individual maneuver or guidance planes defined by each missile and its target. Algorithms 3 and 4 are also based on "myopic" saddle-point calculations. These latter two algorithms apportion the projection of the instantaneous aircraft acceleration into the individual maneuver planes so as to maximize the minimum of a particular function which is related to the line of sight rate of each missile threat. These latter two algorithms are motivated by the concept of anti-proportional navigation.

Each algorithm has the following properties: i) each requires relatively minimal dynamic and parametric information; ii) each provides capability against an N missile threat; iii) each generates aerodynamically feasible aircraft maneuvers which meet both structural and pilot stress limitations; iv) each is computable using foreseeable hardware; v) each exhibits markovian behavior, i.e., each is restartable from present state information.

Simulation results using each algorithm with generic F-4 and AIM-9 truth models characterized by nonlinear differential equations, including lift, drag, gravity, 3-dimensional point mass dynamics, aircraft load factor and roll rate limits, and missile autopilot dynamics and load factor limits have been presented.

Based on the performance of the four heuristic algorithms against seven

representative multiple missile engagement scenarios (Table 4-2a), where algorithm performance in a given scenario is gauged by the minimum miss distance associated with missiles 1 & 2, one observes that, on a scenario by scenario basis, there is no single algorithm whose performance dominates the remaining three algorithms. However, if we calculate the total number of multiple misses, we observe that the algorithms ranked in decreasing order of performance are: Algorithms 1, 4, 2, & 3.

We believe that the importance of this present study lies more in the development and comparative analysis of real-time algorithms for multiple missile evasion, than in the absolute numerical miss distance results obtained herein. The missile model employed in this study exhibits performance capabilities which make it more effective than an actual missile would be. Improving the missile model's realism will therefore affect the numerical miss distance results in the simulation.

Further research work is currently being pursued in several areas to obtain further insight into the potential for actual implementation of a "future generation" of one or several of these algorithms in an operational setting: i) timing of maneuver initiation in the EEG phase; ii) transition strategies between the EEG and IEG phases; iii) the incorporation of load factor as well as bank angle in the aircraft maneuver strategy; iv) a cross comparison between algorithms as a function of initial scenario geometry and state description; v) a parametric sensitivity analysis for items i - iv.

References

Monographs:

(1969) Bryson, A.E. and Y.C. Ho, Applied Optimal Control, Blaisdell, Waltham, Massachusetts.

(1971) Friedman, A., Differential Games, Wiley-Interscience, New York.

(1977) Garnell, P. and D.J. East, Guided Weapon Control Systems, Pergamon Press, Oxford, England.

(1969) Hermes, H. and J.P. LaSalle, Functional Analysis and Time Optimal Control, Academic Press, New York.

(1965) Isaacs, R., Differential Games, Wiley, New York.

(1969) Luenberger, D.G., Optimization By Vector Space Methods, Wiley, New York.

(1969) Owen, G., Game Theory, Saunders, Philadelphia, Pennsylvania.

Journal Articles and Conference Papers:

(1973) Basar, T. and M. Mintz, "A Multistage Pursuit-Evasion Game That Admits a Gaussian Random Process as a Maximin Control Policy," Stochastics, Vol. 1, pp. 25-69.

(1968) Behn, R. and Y.C. Ho, "On a Class of Linear Stochastic Differential Games," IEEE Transactions on Automatic Control, Vol. AC-13, No. 3.

(1967) Da Cunha, N.O. and E. Polak, "Constrained Minimization Under Vector Valued Criteria in Finite Dimensional Spaces," Journal of Mathematical Analysis and Applications, Vol. 19, pp. 103-124.

(1962) Filippov, A.F., "On Certain Questions in the Theory of Optimal Control," SIAM Journal on Control, Vol. 1, No. 1, pp. 76-84.

(1972) Ho, Y.C., "Comment on the Paper By J. Medanic and M. Andjelic," Journal of Optimization Theory and Applications, Vol. 10, No. 3, pp. 187-189.

(1977) Huling, S.F. and M. Mintz, "Optimal Evasion Strategies Against Multiple Missiles -- Based on Differential Game Theory," Proceedings of the 1977 IEEE Conference on Decision and Control, pp. 480-486.

(1971) Medanic, J. and M. Andjelic, "On a Class of Differential Games Without Saddle-Point Solutions," Journal of Optimization Theory and Applications, Vol. 8, No. 6, pp. 413-430.

(1972a) Medanic, J. and M. Andjelic, "Minimax Solution of the Multiple Target Problem," IEEE Transactions on Automatic Control, Vol. AC-17, No. 5.

(1972b) Medanic, J. and M. Andjelic, "Author's Reply to Y.C. Ho," Journal of Optimization Theory and Applications, Vol. 10, No. 3, pp. 190-191.

References (continued)

Journal Articles and Conference Papers:

(1976) Poulter, R. and G. Anderson, "A Guidance Concept for Air-to-Air Missiles Based on Nonlinear Differential Game Theory," Proceedings of the 1976 NAECON Conference, pp. 605-609.

(1971) Reid, R.W. and S.J. Citron, "On Noninferior Performance Index Vectors," Journal of Optimization Theory and Applications, Vol. 7, No. 1, pp. 11-28.

(1969) Rhodes, I.B. and D.G. Luenberger, "Differential Games With Imperfect State Information," IEEE Transactions on Automatic Control, Vol. AC-14, No. 1.

(1979a) Sheketoff, M.S. and M. Mintz, "Real-Time Computation of Aircraft Evasion Strategies Against a Multiple Missile Threat -- A Game theoretic Approach," Proceedings of the Tenth Annual Pittsburgh Conference on Modeling and Simulation (to appear).

(1979b) Sheketoff, M.S. and M. Mintz, "On the Application of Game Theory to the Design of a Real-Time Algorithm for Multiple Missile Evasion," Proceedings of the 1979 International Conference on Information Sciences and Systems (to appear).

(1958) Sion, M., "On General Minimax Theorems," Pacific Journal of Mathematics, Vol. 8, pp. 171-176.

Theses:

(1979) Huling, S.F., On the Application of Game Theory and Time Series Analysis to Fire Control Systems, Ph.D. Dissertation to be submitted to the University of Pennsylvania.

(1975) Poulter, R., Differential Game Guidance Versus Proportional Navigation for an Air-to-Air Missile, M.S. Thesis submitted to the United States Air Force Air University Air Force Institute of Technology, Report No. GA/MC/75-5.

(1979) Sheketoff, M.S., Real-Time Algorithms for Multiple Missile Evasion: A Game Theoretic Approach, M.S. Thesis to be submitted to the University of Pennsylvania.

Reports:

(1977) Brown, C.M. and D.H. Johnson, "Fire Control Simulation (FICS) User's Manual," The Analytical Sciences Corp., Report No. TR-750-1.

(1975) Carpenter, G. and M. Falco, "Analysis of Aircraft Evasion Strategies in Air-to-Air Missile Effectiveness Models," Grumman Aerospace Corp., Report No. RE-506.

References (continued)

Reports:

(1976) Carpenter, G. and M. Falco, "Supercruise Vulnerability to Surface-to-Air Missile Threats," Grumman Aerospace Corp.

(1976) Shinar, J. and D. Steinberg, "Analysis of Optimal Evasive Maneuvers Based on a Linearized Two-Dimensional Kinematic Model," Technion - Israel Institute of Technology, T.A.E. Report No. 230.

Proposals:

(1977) FAAC Proposal FP402U/2762, "Part I - Technical Proposal for a Missile Launch Envelope Sensitivity Analysis," First Ann Arbor Corp.

(1977) Veda Proposal 13005-77U/Q0105, "Technical Proposal Volume I - Missile Launch Envelope Sensitivity Analysis," Veda Corp.

Technical Manuals:

(1977) USAF Flight Manual, TO 1F-4C-1, (for USAF Series F-4C & F-4D aircraft).

Course Notes:

(1975) Hohwiesner, W., "Principles of Airborne Fire Control," Department of Astronautics and Computer Science, United States Air Force Academy. (These course notes are referred to in Chapter 3 of this present study by (AFA, 1975).)

Appendix A

FORTRAN IV Source Programs

```
      PROGRAM ACDYN
C
C     ACDYN.91 --- MYOPIC, MULTIPLE (TWO) MISSILES
C
C     USES LF=1.0, BA=0.0 FOR CONTROLS FOR FIRST STEP,
C        THEN MANEUVERS TO MAXIMIZE ACCELERATION NORMAL
C        TO THE "GUIDANCE PLANE" FOR SINGLE MISSILE CRITERION
C
C     FOR COMBINATION, USE DO LOOP OF LAMBDA'S, I.E.
C        LAMBDA = 0.0 THRU 1.0 BY 0.05, AT EACH STEP,
C        TO FIND THE MIN(LAMBDA) OF THE MAX(UO) OF :
C
C          |(LAMBDA*ACCEL(DOT)G1)+((1-LAMBDA)*ACCEL(DOT)G2)|
C
C        WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
C        TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
C
C     DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
C        FUNCTION OF LAMBDA; I.E.
C
C            TA1(LAMBDA) = DMIS1(FINAL)
C            TA2(LAMBDA) = DMIS2(FINAL)
C
C     ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
C
C     WRITES TO THE TERMINAL, THEN PRINTS
C
C     RK11 = RK12 = 4.5
C
C     STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
C
C     PLOTTED VARIABLES ARE ---
C
C          PL(,1) = USTAR (DEG)
C          PL(,2) = PERFSTAR
C          PL(,3) = USTAR2 (DEG)
C          PL(,4) = PERF2
C          PL(,5) = GX  )
C          PL(,6) = GY  ) COMBINED USING LAMBDA
C          PL(,7) = GZ  )
C          PL(,8) = LF1 (COMMANDED)
C          PL(,9) = SPECIFIC ENERGY (A/C)
C          PL(,10) = SP. EN. (MISSILE 1)
C          PL(,11) = ALPHA (A/C)
C          PL(,12) = GAMMA (A/C)
C          PL(,13) = SIGMA (A/C)
C          PL(,14) = NORM1(G) = (GX1**2+GY1**2+GZ1**2)**0.5
C          PL(,15) = Z (A/C)
C          PL(,16) = AIRSPEED (A/C)
C          PL(,17) = SY1 (L.O.S. PITCH)
C          PL(,18) = THETA1 (L.O.S. YAW)
C          PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
C          PL(,20) = DRAG (A/C)
C          PL(,21) = Z (MISSILE 1)
C          PL(,22) = X (A/C)
C          PL(,23) = X (MISSILE 1)
C          PL(,24) = Y (A/C)
C          PL(,25) = Y (MISSILE 1)
C          PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2
C          PL(,27) = SYDT1
```

```
C          PL(,28) = THEDT1
C          PL(,29) = NORM1(LOSDOT) = (SYDT1**2+THEDT1**2)**0.5
C          PL(,30) = GX1   )
C          PL(,31) = GY1   )  MISSILE 1
C          PL(,32) = GZ1   )
C          PL(,33) = NORM2
C          PL(,34) = GX2   )
C          PL(,35) = GY2   )  MISSILE 2
C          PL(,36) = GZ2   )
C          PL(,37) = NORM2(LOS-DOT)
C          PL(,38) = LF2C
C          PL(,39) = SP. EN. (MISSILE 2)
C          PL(,40) = SY2 (LOS PITCH)
C          PL(,41) = THETA2 (LOS YAW)
C          PL(,42) = X (MISSILE 2)
C          PL(,43) = Y (MISSILE 2)
C          PL(,44) = Z (MISSILE 2)
C          PL(,45) = SYDT2
C          PL(,46) = THEDT2
C          PL(,47) = LAMDA1 * 100.
C          PL(,48) = GSY1
C          PL(,49) = GTH1
C          PL(,50) = GSY2
C          PL(,51) = GTH2
C          PL(,52) = GSY )
C          PL(,53) = GTH )  COMBINED USING LAMBDA
C
      REAL*8 STRING,SDATE
      INTEGER CUTIME
      LOGICAL PRTED,LONCE
      COMMON /PARM7/ LONCE
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
C
      IDUM=CUTIME(0)
      CALL TIMEOD(STRING)
      CALL DATE(SDATE)
      WRITE(6,66) SDATE, STRING
   66 FORMAT(15X,40('*'),' ACDYN.91 ',40('*'),/,
     1    51X,A8,2X,A8,/,1H1)
C
      LONCE=.FALSE.
  999 CALL INIT
      CALL VALUE
      CALL PLOUT(2)
      CALL PLOUT(6)
C
      WRITE(2,100)
  100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 '
     1    'FOR NO INIT PRINT')
      CALL BELL(1)
      READ(1,110) ISEC
  110 FORMAT(I1)
      IF(ISEC.NE.0) GO TO 999
      STOP
      END
```

```
      SUBROUTINE INIT
      DIMENSION IONOF(2)
      REAL MO, LF, LO, M1, LF1, L1, LMIN, LMAX, MAXLF, NORMAX,
     1  LOSMAX, LF2, L2
      INTEGER CUTIME
      LOGICAL PRTED,LONCE
      COMMON /AMP/ TO,Q0S0,ALFA0,CL0,D0,L0,G,MO,C01,C02,C03,
     1  RHO,BETA,CLAF0,S0,D01,D02,M1,CLAF1,S1,D11,RLMAX,
     2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALMEAS
     3  ,PI,TAU,TAFTB,TFLAG,Q2S2,ALFA2,CL2,D2,L2
      COMMON /PARM1/ JJ,KSTEP
      COMMON /PARM2/ VTH0,VTH1,DTH,TSTEP,ICPUTM
      COMMON /PARM3/ N0,I0
      COMMON /PARM5/ MANUVR
      COMMON /PARM7/ LONCE
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,U0,MAXLF
      COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
      COMMON /ARRAY1/ X0(18),XDIN(18)
      COMMON /PLVR/ PL(100,33),PMAX,PMIN,LMAX,LMIN,SPMAX,
     1   SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
C
      DATA IYES/'Y'/, IONOF/'OFF','ON '/
C
    1 FORMAT(I1)
    3 FORMAT(I3)
    6 FORMAT(F12.6)
   11 FORMAT(A1)
C
      IF(LONCE) GO TO 800
      PI=3.14159
      N0=18
      N1=6
      I0=100
      ISEC=1
      RLMAX=600.0
      ALMEAS=25.0*PI/180.
      MU=1243.0
      BETA=0.000030575
      G=32.18
      RHO=0.0023769
      TAFTB=13000.
      C02=-0.7018
      C03=18.141
      CLAF0=5.8986
      S0=530.0
      D01=0.01675
      D02=0.223
      M1=3.20
      CLAF1=22.918
      S1=0.223
      D11=0.7
      D12=0.042
      RK11=4.5
      RK12=4.5
      THRESH=0.1
      VTH0=200.0
      VTH1=200.0
      DTH=15.0
```

```
C
C     BEGIN INPUTTING DATA ---
C       USE FILE 10
C
      READ (10,3) KSTEP
      TSTEP=1.0/FLOAT(KSTEP)
      READ (10,6) XO(1)
      READ (10,6) XO(2)
      READ (10,6) XO(3)
      READ (10,6) XO(4)
      READ (10,6) DA1
      XO(5)=DA1*PI/180.
      READ (10,6) DA2
      XO(6)=DA2*PI/180.
      READ (10,6) TAU
      READ (10,6) XO(7)
      READ (10,6) XO(8)
      READ (10,6) XO(9)
      READ (10,6) XO(10)
      READ (10,6) DA3
      XO(11)=DA3*PI/180.
      READ (10,6) DA4
      XO(12)=DA4*PI/180.
      READ (10,6) XO(13)
      READ (10,6) XO(14)
      READ (10,6) XO(15)
      READ (10,6) XO(16)
      READ (10,6) DA5
      XO(17)=DA5*PI/180.
      READ (10,6) DA6
      XO(18)=DA6*PI/180.
C
      DO 750 I=1,NO
      XOIN(I)=XO(I)
  750 CONTINUE
      GO TO 815
C
  800 CONTINUE
      WRITE(6,813)
  813 FORMAT(1H1)
  815 CONTINUE
      DO 850 I=1,NO
      XO(I)=XOIN(I)
  850 CONTINUE
      IF(ISEC.EQ.2) GO TO 870
C
      LONCE=.TRUE.
      WRITE(2,852) TSTEP, TAU
      WRITE(6,852) TSTEP, TAU
  852 FORMAT(5X,'TSTEP = ',F8.4,', TAU = ',F8.4,/)
      WRITE(2,854)
  854 FORMAT(/,'     XO(.)      AIRCRAFT              MISSILE #1',
     1 '             MISSILE #2',/)
C
      DO 860 I=1,N1
      J=I+6
      K=I+12
      TEMP=XO(I)
      TEMP1=XO(J)
      TEMP2=XO(K)
```

```
      IF(I.LT.5) GO TO 855
      TEMP=TEMP*180./PI
      TEMP1=TEMP1*180./PI
      TEMP2=TEMP2*180./PI
  855 CONTINUE
      WRITE(2,856) I, TEMP, TEMP1, TEMP2
      WRITE(6,858) I, TEMP, J, TEMP1, K, TEMP2
  856 FORMAT(I6,3(5X,G15.6))
  858 FORMAT(3X,'INIT X0(',I1,') = ',G15.6,
     1   1X,'INIT X0(',I2,') = ',G15.6,1X,
     2   'INIT X0(',I2,') = ',G15.6)
  860 CONTINUE
C
  870 CONTINUE
      ICPUTM=CUTIME(0)
      LF1=.25*DA3
      LF2=.25*DA5
      U1=0.0
      U2=0.0
      U0=0.0
      ACTI(1)=0.0
      ACLF(1)=1.0
      ACBA(1)=0.0
      MANUVR=0
      PRTED=.FALSE.
      C01=22345.7
C
      ZD1=X0(3)-X0(9)
      YD1=X0(2)-X0(8)
      XD1=X0(1)-X0(7)
      RXY=SQRT(XD1*XD1+YD1*YD1)
      PSY1=ATAN2(ZD1,RXY)
      THETA1=ATAN2(YD1,XD1)
      DMIS1=SQRT(RXY**2+ZD1**2)
C
      ZD2=X0(3)-X0(15)
      YD2=X0(2)-X0(14)
      XD2=X0(1)-X0(13)
      RXY=SQRT(XD2*XD2+YD2*YD2)
      PSY2=ATAN2(ZD2,RXY)
      THETA2=ATAN2(YD2,XD2)
      DMIS2=SQRT(RXY**2+ZD2**2)
C
      DO 880 I=1,53
      DO 880 J=1,10
      PL(J,I)=0.0
  880 CONTINUE
      PMAX=160000.0
      PMIN=0.0
      LMAX=0.0
      LMIN=6.0
      SPMAX=0.0
      SPMIN=X0(3)+0.5*X0(4)*X0(4)/G
      ALFMAX=0.0
      ALFMIN=0.0
      NORMAX=0.0
      ANMAX=0.0
      ZMAX=0.0
      ZMIN=AMIN1(X0(3),X0(9),X0(15))
      XMAX=0.0
```

```
      XMIN=AMIN1(XO(1),XO(7),XO(13))
      YMAX=0.0
      YMIN=AMIN1(XO(2),XO(8),XO(14))
      VMAX=XO(4)
      VMIN=XO(4)
      DMAX=0.0
      DMIN=1.0E10
      LOSMAX=0.0
C
      WRITE(2,903)
  903 FORMAT(/,'    BEGIN ALL MANEUVERS AFTER 1 STEP')
C
      WRITE(2,913) RK11,RK12
      WRITE(6,913) RK11,RK12
  913 FORMAT(20X,'PROPORTIONAL NAVIGATION GAINS:',/,
     1  12X,'PITCH (RK11) = ',G12.3,', YAW (RK12) = ',G12.3,/)
C
      WRITE(2,916)
  916 FORMAT('  ENTER MAXIMUM LOAD FACTOR FOR A/C (F12.6)')
      CALL BELL(1)
      READ(1,6) MAXLF
      IF(MAXLF.EQ.0.0) MAXLF=6.0
      WRITE(2,918) MAXLF
      WRITE(6,918) MAXLF
  918 FORMAT(10X,'  A/C MAXIMUM LOAD FACTOR = ',F5.2,/)
C
      WRITE(2,926)
  926 FORMAT('  USE AFTERBURNERS IN MANEUVER (Y OR N)?')
      CALL BELL(1)
      READ(1,11) IDUM
      TFLAG=0.C
      IF(IDUM.EQ.IYES) TFLAG=1.0
      I=TFLAG+1.0
      WRITE(2,928) IONOF(I)
      WRITE(6,928) IONOF(I)
  928 FORMAT(10X,'   IN MANEUVER, AFTERBURNERS WILL BE ',A3,/)
C
      ISEC=1
      WRITE(2,971)
  971 FORMAT('  ENTER 1 TO ABORT')
      CALL BELL(1)
      READ(1,1) LOGIC
      IF(LOGIC.EQ.1) STOP
C
      RETURN
      END
```

```
      SUBROUTINE VALUE
      REAL MU, LF, LO, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
     1   NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
     2   NORMG
      DIMENSION ACDF(100,2)
      LOGICAL PRTED, FLAG1, FLAG2
      INTEGER CUTIME
      COMMON /AMP/ T0,Q0S0,ALFA0,CL0,D0,L0,G,MU,C01,C02,C03,
     1  RHO,BETA,CLAF0,S0,D01,D02,M1,CLAF1,S1,D11,RLMAX,
     2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALREAS
     3  ,P1,TAU,TAFTB,TFLAG,Q2S2,ALFA2,CL2,D2,L2
      COMMON /PARM1/ JJ,KSTEP
      COMMON /PARM2/ VTH0,VTH1,DTH,TSTEP,ICPUTM
      COMMON /PARM3/ NO,IO
      COMMON /PARM4/ STEP
      COMMON /PARM5/ MANUVR
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
      COMMON /ARRAY1/ X0(18),X0IN(18)
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,U0,MAXLF
      COMMON /FLVR/ PL(100,53),PMAX,PMIN,LMAX,LMIN,SPMAX,
     1   SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
      COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
C
      EQUIVALENCE (GAMA,X0(5)),(SIGMA,X0(6))
      EQUIVALENCE (V0,X0(4)),(V1,X0(10)),(V2,X0(16))
C
      DATA TESTN /1.0/, IEXIT/'X'/, IQUIK/'Q'/
C
      PERF(U)=COEFA*COS(U)+COEFB*SIN(U)+COEFC
C
      COSGAM=COS(GAMA)
      SINGAM=SIN(GAMA)
      COSSIG=COS(SIGMA)
      SINSIG=SIN(SIGMA)
C
      TA=0.0
      JJ=0
      DUM1=1.0E10
      DUM2=1.0E10
      FLAG1=.FALSE.
      FLAG2=.FALSE.
      LAMDA1=0.5
C
C     FLAG(I) INDICATES THAT MISSILE I ALREADY GOT AWAY
C
  100 CONTINUE
      IF(FLAG1) LAMDA1=0.0
      IF(FLAG2) LAMDA1=1.0
      LAMDA2=1.0-LAMDA1
      IVAL=50
      TPRINT=TA
      V0X=V0*COSGAM*COSSIG
      V0Y=V0*COSGAM*SINSIG
      V0Z=V0*SINGAM
      V1X=V1*COS(X0(11))*COS(X0(12))
      V1Y=V1*COS(X0(11))*SIN(X0(12))
      V1Z=V1*SIN(X0(11))
      V2X=V2*COS(X0(17))*COS(X0(18))
      V2Y=V2*COS(X0(17))*SIN(X0(18))
```

```
      V2Z=V2*SIN(XO(17))
      VRELX1=VCX-V1X
      VRELY1=VCY-V1Y
      VRELZ1=VCZ-V1Z
      VRELT1=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
      VRELX2=VCX-V2X
      VRELY2=VCY-V2Y
      VRELZ2=VCZ-V2Z
      VRELT2=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
      DELX1=XO(1)-XO(7)
      DELY1=XO(2)-XO(8)
      DELZ1=XO(3)-XO(9)
      DELX2=XO(1)-XO(13)
      DELY2=XO(2)-XO(14)
      DELZ2=XO(3)-XO(15)
C
      DUM1=AMIN1(DUM1,DMIS1)
      DUM2=AMIN1(DUM2,DMIS2)
      DMIS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
      DMIS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
      DSV1=0.5*DMIS1/VRELT1
      DSV2=0.5*DMIS2/VRELT2
      STEP=AMIN1(TSTEP,DSV1,DSV2)
      DMIS=AMIN1(DMIS1,DMIS2)
      IF(JJ.GT.0) MANUVR=1
      LF=1.0
      ULAST=UO
      UO=0.0
      IF(MANUVR.NE.1) GO TO 300
C
      UO=ULAST
      LF=MAXLF
      ISEC=ISEC+1
      GX1 = VRELY1*DELZ1 - VRELZ1*DELY1
      GY1 = VRELZ1*DELX1 - VRELX1*DELZ1
      GZ1 = VRELX1*DELY1 - VRELY1*DELX1
      NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
      GX2 = VRELY2*DELZ2 - VRELZ2*DELY2
      GY2 = VRELZ2*DELX2 - VRELX2*DELZ2
      GZ2 = VRELX2*DELY2 - VRELY2*DELX2
      NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
      IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 200
C
C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
C
      LF=ACLF(ISEC-1)
      GO TO 290
C
  200 CONTINUE
      GX1=GX1/NORM1
      GY1=GY1/NORM1
      GZ1=GZ1/NORM1
      GX2=GX2/NORM2
      GY2=GY2/NORM2
      GZ2=GZ2/NORM2
      PERFM=1.0E20
C
      DO 270 LAMB=1,21
      LAMDA1=FLOAT(LAMB-1)/20.
      LAMDA2=1.0-LAMDA1
```

```
      IF(FLAG1.AND.(LAMB.GT.1)) GO TO 280
      IF(FLAG2.AND.(LAMB.LT.21)) GO TO 270
C
      GX=LAMDA1*GX1+LAMDA2*GX2
      GY=LAMDA1*GY1+LAMDA2*GY2
      GZ=LAMDA1*GZ1+LAMDA2*GZ2
C
      COEFA=GZ*COSGAM-GY*SINGAM*SINSIG-GX*SINGAM*COSSIG
      COEFA=COEFA*(LO+TO*SIN(ALFAO))
      COEFB=GY*COSGAM-GX*SINGAM
      COEFB=COEFB*(LO+TO*SIN(ALFAO))
      COEFC=((TO*COS(ALFAO)-DO)*(GZ*SINGAM+GY*COSGAM*SINSIG
     1   +GX*COSGAM*COSSIG)) - GZ*MO*G
C
      USTAR=ATAN2(COEFB,COEFA)
      USTAR2=USTAR+PI
      IF(USTAR.GT.0.0) USTAR2=USTAR-PI
  220 CONTINUE
      PERFST=PERF(USTAR)
      PERF2=PERF(USTAR2)
      ABST=ABS(PERFST)
      ABS2=ABS(PERF2)
C
C     USE "BEST" U AND RATE LIMIT TO RLMAX
C
      IFLAG=IVAL
      UMAX=USTAR
      PERMIN=ABST
      IF(ABST.GE.ABS2) GO TO 230
      UMAX=USTAR2
      IFLAG=-IVAL
      PERMIN=ABS2
  230 CONTINUE
      ROLL=(UMAX-UO)*180./PI
      IF(ROLL.GT.180.) ROLL=ROLL-360.
      IF(ROLL.LT.(-180.)) ROLL=ROLL+360.
      AROLL=ABS(ROLL)
C
C     WE NOW RATE-LIMIT WHICHEVER ANGLE WE GET TO RLMAX
C          DEG PER SEC...
C
      IF(AROLL.LE.(RLMAX*STEP)) GO TO 250
      ROLL=ROLL*RLMAX*STEP/AROLL
      IF(AROLL.LE.175.) GO TO 250
C
C     ROLL IS ESSENTIALLY A COMPLETE FLIP --- USE PLUS/MINUS
C         RLMAX INTO PERF FUNCTION TO CHECK FOR BEST WAY
C         TO MAKE FLIPS
C
      IVAL=75
      USTAR=UO+(RLMAX*STEP*PI/180.)
      IF(USTAR.GT.PI) USTAR=USTAR-(2.0*PI)
      USTAR2=UO-(RLMAX*STEP*PI/180.)
      IF(USTAR2.LE.(-PI)) USTAR2=USTAR2+(2.0*PI)
      GO TO 220
C
  250 CONTINUE
      IF(PERMIN.GE.PERFM) GO TO 270
      PERFM=PERMIN
      JMIN=LAMB
```

```
      RLMIN=ROLL
      PL(ISEC,1)=USTAR*180./PI
      PL(ISEC,2)=PERFST
      PL(ISEC,3)=USTAR2*180./PI
      PL(ISEC,4)=PERF2
      PL(ISEC,5)=GX
      PL(ISEC,6)=GY
      PL(ISEC,7)=GZ
      PL(ISEC,26)=IFLAG
      PL(ISEC,47)=5-(JMIN-1)
      PMAX=AMAX1(PMAX,PERFST,PERF2)
      PMIN=AMIN1(PMIN,PERFST,PERF2)
  270 CONTINUE
  280 CONTINUE
C
      ROLL=RLMIN
      UO=UO+(ROLL*PI/180.)
      IF(UO.GT.PI) UO=UO-(2.0*PI)
      IF(UO.LE.(-PI)) UO=UO+(2.0*PI)
C
C     CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
C
      ACCX=((TO*COS(ALFAO)-DO)*(COSGAM*COSSIG))-
     1   ((LO+TO*SIN(ALFAO))*(SIN(UO)*SINGAM+COS(UO)*
     2   SINGAM*COSSIG))
      ACCY=((TO*COS(ALFAO)-DO)*(COSGAM*SINSIG))+
     1   ((LO+TO*SIN(ALFAO))*(SIN(UO)*COSGAM-COS(UO)*
     2   SINGAM*SINSIG))
      ACCZ=((TO*COS(ALFAO)-DO)*SINGAM)-(MO*G)+
     1   ((LO+TO*SIN(ALFAO))*COSGAM*COS(UO))
      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
C
C     STORE PLOTTED VARIABLES
C
      PL(ISEC,14)=NORM1
      PL(ISEC,19)=ANORM/MO
      PL(ISEC,30)=GX1
      PL(ISEC,31)=GY1
      PL(ISEC,32)=GZ1
      PL(ISEC,33)=NORM2
      PL(ISEC,34)=GX2
      PL(ISEC,35)=GY2
      PL(ISEC,36)=GZ2
      NORMAX=AMAX1(NORMAX,NORM1,NORM2)
      ANMAX=AMAX1(ANMAX,ANORM)
C
  290 CONTINUE
      IF(PRTED) GO TO 300
      PRTED=.TRUE.
      CO1=CO1+TAFTB*TFLAG
      TCHG=TPRINT
      WRITE(2,310) TPRINT
      WRITE(6,310) TPRINT
  310 FORMAT(2X,'START MANEUVER AT T = ',F6.2)
      CALL BELL(3)
C
      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
      ALFMAX=ALFAO
      ALFMIN=ALFAO
```

```
      ZMAX=AMIN1(XC(3),XC(9),XC(15))
      ZMIN=AMAX1(XC(3),XC(9),XC(15))
      XMAX=AMAX1(XC(1),XC(7),XC(13))
      XMIN=AMIN1(XC(1),XC(7),XC(13))
      YMAX=AMAX1(XC(2),XC(8),XC(14))
      YMIN=AMIN1(XC(2),XC(8),XC(14))
      VMAX=V0
      VMIN=V0
      DMAX=D0
      DMIN=D0
      LOSMAX=0.0
C
  300 CONTINUE
      ACBA(ISEC)=U0*180./PI
      ACLF(ISEC)=LF
      ACTI(ISEC)=TPRINT
      ACDM(ISEC,1)=DMIS1
      ACDM(ISEC,2)=DMIS2
C
      IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
      WRITE (2,400) TPRINT, DMIS1, DMIS2
      WRITE (6,400) TPRINT, DMIS1, DMIS2
  400 FORMAT (5X,'TIME = ',F10.3,2X,'DSEP1 = ',G12.3,2X,
     1    'DSEP2 = ',G12.3)
C
  477 CONTINUE
      IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
      IF(DMIS.LT.DTH) GO TO 520
      IF(TA.LT.3.0) GO TO 480
      FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
      FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
      IF(FLAG1.AND.FLAG2) GO TO 540
  480 CONTINUE
      CALL INTGOX
      TA=TA+STEP
      JJ=JJ+1
      IF(ISEC.GE.10) GO TO 530
      GO TO 100
C
  510 CONTINUE
      WRITE (2,515) TPRINT
      WRITE (6,515) TPRINT
  515 FORMAT(1X,'**   A/C OR MISSILE VEL. IS TOO LOW ',
     1 'AT TIME: ',F10.3,'    **',/)
      GO TO 600
C
C     HIT OCCURRED, PRINT OUT
  520 CONTINUE
      WRITE(2,525) TPRINT,DUM1,DUM2
      WRITE(6,525) TPRINT,DUM1,DUM2
  525 FORMAT(1X,'*****  HIT  AT TIME = ',F10.3,
     1   '  *****',/,5X,'BEST DSEPS WERE 1:',
     2   G15.6,', & 2:',G15.6,/)
      GO TO 600
C
  530 CONTINUE
      WRITE(2,535) TPRINT
      WRITE(6,535) TPRINT
  535 FORMAT(5X,'TIME LIMIT AT T = ',F6.2)
      GO TO 600
```

```
C
C     CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
  540 CONTINUE
      WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
      WRITE(6,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
  544 FORMAT(3X,'**** CLOSURE RATE NEGATIVE AT TIME = ',F10.3,
     1    ' ****',/,5X,'TA1 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/,
     2    5X,'TA2 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/)
      GO TO 600
C
  600 CONTINUE
      CALL BELL(1)
C
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      N10=6
      DO 620 J=1,N10
      JJJ=J+6
      KK=J+12
      TEMP=XU(J)
      TEMP1=XU(JJJ)
      TEMP2=XU(KK)
      IF(J.LT.5) GO TO 605
      TEMP=TEMP*180./PI
      TEMP1=TEMP1*180./PI
      TEMP2=TEMP2*180./PI
  605 CONTINUE
      WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
      WRITE(6,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
  610 FORMAT(2X,'XU(',I1,'): ',G13.4,4X,
     1    'XU(',I2,'): ',G13.4,4X,
     2    'XU(',I2,'): ',G13.4)
  620 CONTINUE
C
      WRITE (2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
      WRITE (6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
  625 FORMAT(/,10X,'DELX1: ',G12.3,2X,'DELY1: ',G12.3,/,
     1  10X,'DELZ1: ',G12.3,2X,'DMIS1: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      WRITE (2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
      WRITE (6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
  630 FORMAT(/,10X,'DELX2: ',G12.3,2X,'DELY2: ',G12.3,/,
     1  10X,'DELZ2: ',G12.3,2X,'DMIS2: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      CALL BELL(1)
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
  635 FORMAT(A1)
C
      WRITE(6,631)
  631 FORMAT(1H1)
      DO 650 I=1,ISEC
      IF(MOD(I,20).EQ.0) CALL BELL(1)
      IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
      WRITE(2,636) ACTI(I),ACBA(I),PL(I,6),PL(I,38)
  636 FORMAT(1X,'TIME ',F5.2,3X,'ACBA ',F6.2,3X,'LF1C ',F8.1,
     1    3X,'LF2C ',F8.1)
```

```
  650 CONTINUE
  660 CONTINUE
      DO 670 I=1,ISEC
      WRITE(6,637) ACTI(I),ACLF(I),ACBA(I),PL(I,8),PL(I,38),
     1   PL(I,19),PL(I,14),PL(I,11),PL(I,20),
     2   (ACUM(I,J),J=1,2),PL(I,47)
  637 FORMAT(1X,'TIME = ',0PF6.2,3X,'ACLF = ',F5.1,3X,
     1 'ACBA(DEG) = ',F8.2,3X,'LFTC = ',F8.1,3X,
     2   'LFEC = ',F8.2,3X,'NORM(A) = ',G11.2,/,20X,
     3   'NORM1(G) = ',G11.2,3X,'ALPHA = ',F6.2,3X,
     4   'DRAG = ',G11.2,/,20X,'DMIS1 = TA(1) = ',
     5   F9.2,5X,'DMIS2 = TA(2) = ',F9.2,5X,
     6   'LAMBDA1 = ',-2PF5.2,/)
  670 CONTINUE
C
      IDUM=CUTIME(0)-ICPUTM
      WRITE(2,639) IDUM
      WRITE(6,639) IDUM
  639 FORMAT(4X,'CPU TIME (SECONDS) = ',I5,0PF5.0)
C
      RETURN
      END
```

```
      SUBROUTINE INTBOX
      REAL MO, LF, LO, M1, LF1, L1, LMAX, LMIN, MAXLF, NORMAX,
     1  LOSMAX, LF2, L2, MVO, MV1, MV2
      LOGICAL PRTED
      DIMENSION XPO(15)
      COMMON /AMP/ TO,QOSO,ALFAO,CLO,DO,LO,G,MO,C01,C02,C03,
     1  RHO,BETA,CLAFO,SO,D01,D02,M1,CLAF1,S1,D11,RLMAX,
     2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALREAS
     3  ,P1,TAU,TAFTB,TFLAG,Q2S2,ALFA2,CL2,D2,L2
      COMMON /ARRAY1/ XO(18),XOIN(18)
      COMMON /PARM1/ JJ,KSTEP
      COMMON /PARM3/ NO,IO
      COMMON /PARM4/ STEP
      COMMON /PARM5/ MANUVR
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
      COMMON /PLVR/ PL(100,55),PMAX,PMIN,LMAX,LMIN,SPMAX,
     1   SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
      COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
C
      EQUIVALENCE (GAMA,XO(5)),(SIGMA,XO(6))
      EQUIVALENCE (VO,XO(4)),(V1,XO(10)),(V2,XO(16))
C
C     FIRST CALCULATE AIRCRAFT DYNAMICS
C
      COSGAM=COS(GAMA)
      SINGAM=SIN(GAMA)
      COSSIG=COS(SIGMA)
      SINSIG=SIN(SIGMA)
C
      MVO=MO*VO
      TO=C01+C02*XO(3)+C03*VO
      QOSO=0.5*RHO*EXP(-BETA*XO(3))*VO*VO*SO
C
C     LIMIT ALFAO TO STALL ANGLE, AND STORE ALPHA ACHIEVED
C
      ALFAO=AMIN1((MO*G*LF/(CLAFO*QOSO)),ALREAS)
      ACLF(ISEC)=ALFAO*CLAFO*QOSO/(MO*G)
      SINALF=SIN(ALFAO)*TO
      CLO=CLAFO*ALFAO
      DO=QOSO*(D01+D02*CLO**2)
      LO=CLO*QOSO
      XPO(1)=VO*COSGAM*COSSIG
      XPO(2)=VO*COSGAM*SINSIG
      XPO(3)=VO*SINGAM
      XPO(4)=(TO*COS(ALFAO)-DO)/MO - G*SINGAM
      XPO(5)=(LO+SINALF)*COS(UO)/(MVO)-G*COSGAM/VO
      XPO(6)=(LO+SINALF)*SIN(UO)/(MVO*COSGAM)
C
C     MISSILE # 1 --- CALCULATE DYNAMICS AND CONTROLS
C
      COSGAM=COS(XO(11))
      SINGAM=SIN(XO(11))
      COSSIG=COS(XO(12))
      SINSIG=SIN(XO(12))
C
      MV1=M1*V1
      Q1S1=0.5*RHO*EXP(-BETA*XO(9))*V1*V1*S1
      ALFA1C=M1*G*LF1/(CLAF1*Q1S1)
```

```
      IF(JJ.EQ.0) ALFA1=ALFA1C
      IF(TAU.GE.0.00001)ALFA1=EXP(-STEP/TAU)*(ALFA1-ALFA1C) + ALFA1C
      IF(TAU.LT.0.00001) ALFA1=ALFA1C
      CL1=CLAF1*ALFA1
      D1=Q1S1*(D11+D12*CL1**2)
      L1=CL1*Q1S1
      XPO(7)=V1*COSGAM*COSSIG
      XPO(8)=V1*COSGAM*SINSIG
      XPO(9)=V1*SINGAM
      XPO(10)=-D1/M1 - G*SINGAM
      XPO(11)=L1*COS(U1)/(MV1) - G*COSGAM/V1
      XPO(12)=L1*SIN(U1)/(MV1*COSGAM)
      XD1=XO(1)-XO(7)
      YD1=XO(2)-XO(8)
      ZD1=XO(3)-XO(9)
      XPD1=XPO(1)-XPO(7)
      YPD1=XPO(2)-XPO(8)
      ZPD1=XPO(3)-XPO(9)
      RR1=XD1**2+YD1**2+ZD1**2
      RRXY1=XD1**2+YD1**2
      RXY1=SQRT(RRXY1)
      PSY1=ATAN2(ZD1,RXY1)
      THETA1=ATAN2(YD1,XD1)
      THEDT1=(XD1*YPD1-YD1*XPD1)/RRXY1
      SYDT1=(RXY1*ZPD1-ZD1*(XD1*XPD1+YD1*YPD1)/RXY1)/RR1
      DTNORM=(SYDT1**2+THEDT1**2)**0.5
      ARG1=RK12*THEDT1/(RK11*SYDT1+G*COSGAM/V1)
      U1=ATAN(ARG1)
      TEST=ABS(THEDT1)
      IF(TEST .GE. THRESH) LF1=
     1 RK12*THEDT1*V1*COSGAM/(G*SIN(U1))
      IF(TEST .LT. THRESH) LF1=
     1 (RK11*SYDT1+G*COSGAM/V1)*(V1/(G*COS(U1)))
C
      PL(ISEC,8)=LF1
      LMAX=AMAX1(LMAX,ABS(LF1))
      LMIN=AMIN1(LMIN,(LF1))
      LF1=AMIN1(15.0,AMAX1(-15.0,LF1))
C
C     MISSILE # 2 --- CALCULATE DYNAMICS AND CONTROLS
C
      COSGAM=COS(XO(17))
      SINGAM=SIN(XO(17))
      COSSIG=COS(XO(18))
      SINSIG=SIN(XO(18))
C
      MV2=M1*V2
      Q2S2=0.5*RHO*EXP(-BETA*XO(15))*V2*V2*S1
      ALFA2C=M1*G*LF2/(CLAF1*Q2S2)
      IF(JJ.EQ.0) ALFA2=ALFA2C
      IF(TAU.GE.0.00001)ALFA2=EXP(-STEP/TAU)*(ALFA2-ALFA2C) + ALFA2C
      IF(TAU.LT.0.00001) ALFA2=ALFA2C
      CL2=CLAF1*ALFA2
      D2=Q2S2*(D11+D12*CL2**2)
      L2=CL2*Q2S2
      XPO(13)=V2*COSGAM*COSSIG
      XPO(14)=V2*COSGAM*SINSIG
      XPO(15)=V2*SINGAM
      XPO(16)=-D2/M1 - G*SINGAM
      XPO(17)=L2*COS(U2)/(MV2) - G*COSGAM/V2
```

```
      XPO(19)=L2*SIN(U2)/(MV2*COSGAM)
      XD2=XO(1)-XO(13)
      YD2=XO(2)-XO(14)
      ZD2=XO(3)-XO(15)
      XPD2=XPO(1)-XPO(13)
      YPD2=XPO(2)-XPO(14)
      ZPD2=XPO(3)-XPO(15)
      RR2=XD2**2+YD2**2+ZD2**2
      RRXY2=XD2**2+YD2**2
      RXY2=SQRT(RRXY2)
      PSY2=ATAN2(ZD2,RXY2)
      THETA2=ATAN2(YD2,XD2)
      THEDT2=(XD2*YPD2-YD2*XPD2)/RRXY2
      SYDT2=(RXY2*ZPD2-ZD2*(XD2*XPD2+YD2*YPD2)/RXY2)/RR2
      DTN2=(SYDT2**2+THEDT2**2)**0.5
      ARG2=RK12*THEDT2/(RK11*SYDT2+G*COSGAM/V2)
      U2=ATAN(ARG2)
      TEST=ABS(THEDT2)
      IF(TEST .GE. THRESH) LF2=
     1  RK12*THEDT2*V2*COSGAM/(G*SIN(U2))
      IF(TEST .LT. THRESH) LF2=
     1  (RK11*SYDT2+G*COSGAM/V2)*(V2/(G*COS(U2)))
C
      PL(ISEC,38)=LF2
      LMAX=AMAX1(LMAX,ABS(LF2))
      LMIN=AMIN1(LMIN,(LF2))
      LF2=AMIN1(15.0,AMAX1(-15.0,LF2))
C
C     DO INTEGRATION
C
      DO 100 I=1,NU
      XO(I)=XO(I) + STEP*XPO(I)
  100 CONTINUE
C
C     CALCULATE GUIDANCE PLANES IN SY-THETA COORDINATES
C
      GSY1=-THEDT1*COS(PSY1)
      GTH1=SYDT1
      GSTN=SQRT(GSY1**2+GTH1**2)
      IF(GSTN.GE.(1.0)) GSY1=GSY1/GSTN
      IF(GSTN.GE.(1.0)) GTH1=GTH1/GSTN
C
      GSY2=-THEDT2*COS(PSY2)
      GTH2=SYDT2
      GSTN=SQRT(GSY2**2+GTH2**2)
      IF(GSTN.GE.(1.0)) GSY2=GSY2/GSTN
      IF(GSTN.GE.(1.0)) GTH2=GTH2/GSTN
C
      ALAM=PL(ISEC,47)/100.
      GSY=ALAM*GSY1+(1-ALAM)*GSY2
      GTH=ALAM*GTH1+(1-ALAM)*GTH2
C
C     STORE PLOTTED VARIABLES
C
      SPEN0=XO(3) + 0.5*V0*V0/G
      SPEN1=XO(9) + 0.5*V1*V1/G
      SPEN2=XO(15) + 0.5*V2*V2/G
      PL(ISEC,9)=SPEN0
      PL(ISEC,10)=SPEN1
      PL(ISEC,11)=ALFA0*180./PI
```

```
      PL(ISEC,12)=GAMA*180./PI
      PL(ISEC,13)=SIGMA*180./PI
      PL(ISEC,15)=XO(3)
      PL(ISEC,16)=VO
      PL(ISEC,17)=PSY1*180./PI
      PL(ISEC,18)=THETA1*180./PI
      PL(ISEC,20)=DO
      PL(ISEC,21)=XO(9)
      PL(ISEC,22)=XO(1)
      PL(ISEC,23)=XO(7)
      PL(ISEC,24)=XO(2)
      PL(ISEC,25)=XO(8)
      PL(ISEC,27)=ASPEC(SYDT1)
      PL(ISEC,28)=ASPEC(THEDT1)
      PL(ISEC,29)=ASPEC(DTNORM)
      PL(ISEC,37)=ASPEC(DTN2)
      PL(ISEC,39)=SPEN2
      PL(ISEC,40)=PSY2*180./PI
      PL(ISEC,41)=THETA2*180./PI
      PL(ISEC,42)=XO(13)
      PL(ISEC,43)=XO(14)
      PL(ISEC,44)=XO(15)
      PL(ISEC,45)=ASPEC(SYDT2)
      PL(ISEC,46)=ASPEC(THEDT2)
      PL(ISEC,48)=GSY1
      PL(ISEC,49)=GTH1
      PL(ISEC,50)=GSY2
      PL(ISEC,51)=GTH2
      PL(ISEC,52)=GSY
      PL(ISEC,53)=GTH
C
      SPMAX=AMAX1(SPMAX,SPENO,SPEN1,SPEN2)
      SPMIN=AMIN1(SPMIN,SPENO,SPEN1,SPEN2)
      ALFMAX=AMAX1(ALFAO*180./PI,ALFMAX)
      ALFMIN=AMIN1(ALFAO*180./PI,ALFMIN)
      ZMAX=AMAX1(ZMAX,XO(3),XO(9),XO(15))
      ZMIN=AMIN1(ZMIN,XO(3),XO(9),XO(15))
      XMAX=AMAX1(XMAX,XO(1),XO(7),XO(13))
      XMIN=AMIN1(XMIN,XO(1),XO(7),XO(13))
      YMAX=AMAX1(YMAX,XO(2),XO(8),XO(14))
      YMIN=AMIN1(YMIN,XO(2),XO(8),XO(14))
      VMAX=AMAX1(VMAX,VO)
      VMIN=AMIN1(VMIN,VO)
      DMAX=AMAX1(DMAX,DO)
      DMIN=AMIN1(DMIN,DO)
      LOSMAX=AMAX1(LOSMAX,ASPEC(DTNORM),ASPEC(DTN2))
C
      RETURN
      END
```

```
      SUBROUTINE PLOUT(IO)
      DIMENSION XT(99), YT(98), ARAY1(99), ARAY2(98)
      REAL LMAX, LMIN, MAXLF, NORMAX, LOSMAX, LOSMIN, LF
      INTEGER CUTIME
      LOGICAL PRTED
      COMMON /FLVR/ PL(100,55),PMAX,PMIN,LMAX,LMIN,SPMAX,
     1   SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
      COMMON /PARM2/ VTH0,VTH1,DTH,TSTEP,ICPUTM
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
C
      EQUIVALENCE (XT(1),ACTI(2)),(YT(1),ACTI(3))
C
      DATA IEXIT/'X'/
C
      JSEC=ISEC-1
      KSEC=ISEC-2
      IF(KSEC.LE.1) RETURN
C
      IF(IO.EQ.6) WRITE(2,11)
   11 FORMAT('  TYPE ''X'' FOR NO PRINTED GRAPHS')
C
      CALL BELL(1)
      READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
    1 FORMAT(A1)
C
      PMAX=AMAX1(PMAX,ABS(PMIN))
      PMIN=-PMAX
      APMIN=PMAX
C
      CALL XSPRED(ACTI(1),ACTI(ISEC),'LIN',-180.,180.,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('F',XT,PL(2,26),JSEC)
      CALL PLARRY('*',XT,PL(2,1),JSEC)
      CALL PLARRY('2',XT,PL(2,3),JSEC)
      CALL PLARRY('L',XT,PL(2,47),JSEC)
      CALL PLARRY('U',ACTI,ACBA,ISEC)
      CALL GRAPH('MANEUVERING TIME',16,'BANK ANGLES & LAMBDA'
     1    ,20)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      IF(IO.EQ.6) WRITE(2,12)
   12 FORMAT('  PLEASE BE PATIENT --- I''M GOING AS FAST AS I CAN')
C
      CALL YSPRED(ACTI(1),ACTI(ISEC),'LIN',-180.,180.,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('U',ACTI,ACBA,ISEC)
      CALL GRAPH('MANEUVERING TIME',16,'BANK ANGLES',11)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL YSPRED(ACTI(1),ACTI(ISEC),'LIN',100.,0.,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('L',XT,PL(2,47),JSEC)
```

```
      CALL GRAPH('MANEUVERING TIME',16,'LAMBDA',6)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      DO 60 I=3,ISEC
      ARAY1(I)=ABS(PL(I,2))
      ARAY2(I)=ABS(PL(I,4))
      APMIN=AMIN1(APMIN,ARAY1(I),ARAY2(I))
   60 CONTINUE
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',APMIN,PMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('+',YT,ARAY1,KSEC)
      CALL PLARRY('2',YT,ARAY2,KSEC)
      CALL GRAPH('MANEUVERING TIME',16,'ABS PERFS',9)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      DO 80 I=2,ISEC
      PL(I,1)=PL(I,1)*1.0E3
      PL(I,3)=PL(I,3)*1.0E3
      ACBA(I)=ACBA(I)*1.0E3
   80 CONTINUE
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',PMIN,PMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('P',XT,PL(2,2),JSEC)
      CALL PLARRY('P',XT,PL(2,4),JSEC)
      CALL PLARRY('+',XT,PL(2,1),JSEC)
      CALL PLARRY('2',XT,PL(2,3),JSEC)
      CALL PLARRY('U',ACTI,ACBA,ISEC)
      CALL GRAPH('U=BANK USED;+,2=BANK CALC''D;P,R=PERFS',37,
     1    'BANKS * 1000 AND PERFS',22)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      DO 90 I=2,ISEC
      PL(I,1)=PL(I,1) 1.0E-3
      PL(I,3)=PL(I,3)*1.0E-3
      ACBA(I)=ACBA(I)*1.0E-3
   90 CONTINUE
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',VMIN,VMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('V',ACTI,PL(1,16),ISEC)
      CALL GRAPH('MANEUVERING TIME',16,'AIRSPEED',8)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(XMIN,XMAX,'LIN',YMIN,YMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('1',PL(1,23),PL(1,25),JSEC)
      CALL PLARRY('2',PL(1,42),PL(1,43),JSEC)
      CALL PLARRY('A',PL(1,22),PL(1,24),JSEC)
      CALL GRAPH('X-Y PROJECTION --- X',20,'Y',1)
      IF(IO.EQ.2) READ(1,1) LOGIC
```

```
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(XMIN,XMAX,'LIN',ZMIN,ZMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('1',PL(1,23),PL(1,21),JSEC)
      CALL PLARRY('2',PL(1,42),PL(1,44),JSEC)
      CALL PLARRY('R',PL(1,22),PL(1,15),JSEC)
      CALL GRAPH('X-Z PROJECTION --- X',20,'Z',1)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(YMIN,YMAX,'LIN',ZMIN,ZMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('1',PL(1,23),PL(1,21),JSEC)
      CALL PLARRY('2',PL(1,43),PL(1,44),JSEC)
      CALL PLARRY('R',PL(1,24),PL(1,15),JSEC)
      CALL GRAPH('Y-Z PROJECTION --- Y',20,'Z',1)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      IF(IO.EQ.6) WRITE(2,13)
   13 FORMAT(' I JUST FINISHED PROJECTIONS -- HANG IN THERE')
C
      LOSMIN=-LOSMAX
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',LOSMIN,LOSMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('N',ACTI,PL(1,29),JSEC)
      CALL PLARRY('S',ACTI,PL(1,27),JSEC)
      CALL PLARRY('T',ACTI,PL(1,28),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'LOS RATES AND NORM1',19)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',LOSMIN,LOSMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('N',ACTI,PL(1,37),JSEC)
      CALL PLARRY('S',ACTI,PL(1,45),JSEC)
      CALL PLARRY('T',ACTI,PL(1,46),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'LOS RATES AND NORM2',19)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL YSPRED(ACTI(1),ACTI(ISEC),'LIN',-180.,180.,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('S',ACTI,PL(1,17),JSEC)
      CALL PLARRY('P',ACTI,PL(1,40),JSEC)
      CALL PLARRY('T',ACTI,PL(1,18),JSEC)
      CALL PLARRY('2',ACTI,PL(1,41),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'NAV ANGLES',10)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL YSPRED(ACTI(1),ACTI(ISEC),'LIN',-180.,180.,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
```

```
      CALL PLARRY('S',ACTI,PL(1,13),JSEC)
      CALL PLARRY('C',ACTI,PL(1,12),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'POSITN ANGLES',13)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      IF(IO.EQ.6) WRITE(2,14)
   14 FORMAT(' NEXT ARE THE GUIDANCE PLANES -- NOT MUCH MORE')
C
      CALL XSPRED(XT(1),XT(JSEC),'LIN',-1.0,1.0,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('S',XT,PL(2,52),JSEC)
      CALL PLARRY('T',XT,PL(2,53),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'GUIDANCE PLANE',14)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL XSPRED(XT(1),XT(JSEC),'LIN',-1.0,1.0,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('S',XT,PL(2,48),JSEC)
      CALL PLARRY('T',XT,PL(2,49),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'GUIDANCE PLANE-1',16)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL XSPRED(XT(1),XT(JSEC),'LIN',-1.0,1.0,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('S',XT,PL(2,50),JSEC)
      CALL PLARRY('T',XT,PL(2,51),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'GUIDANCE PLANE-2',16)
C
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',LMIN,LMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('1',ACTI,PL(1,6),JSEC)
      CALL PLARRY('2',ACTI,PL(1,38),JSEC)
      CALL PLARRY('A',ACTI,ACLF,ISEC)
      CALL GRAPH('MANEUVERING TIME',16,'LOAD FACTORS',12)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',ALFMIN,ALFMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('A',ACTI,PL(1,11),ISEC)
      CALL GRAPH('MANEUVERING TIME',16,'ANGLE OF ATTACK',15)
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',DMIN,DMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('D',ACTI,PL(1,20),JSEC)
      CALL GRAPH('MANEUVERING TIME',16,'DRAG',4)
```

```
      IF(IO.EQ.2) READ(1,1) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
C
      CALL SPRED(ACTI(1),ACTI(ISEC),'LIN',SPMIN,SPMAX,'LIN')
      IF(IO.EQ.2) CALL VDT
      CALL BOX
      CALL PLARRY('1',ACTI,PL(1,10),ISEC)
      CALL PLARRY('2',ACTI,PL(1,39),ISEC)
      CALL PLARRY('A',ACTI,PL(1,9),ISEC)
      CALL GRAPH('MANEUVERING TIME',16,'SPECIFIC ENERGIES',17)
      IF(IO.EQ.6) WRITE(2,15)
   15 FORMAT('    DONE !!!!!!')
C
      IDUM=CUTIME(0)-ICPUTM
      WRITE(10,239) IDUM
  239 FORMAT(5X,'CPU TIME (SECONDS) = ',I5)
C
      RETURN
      END
```

```
      SUBROUTINE BELL(N)
      INTEGER*2 A(10)
      DATA A/10*ZE000/
  100 FORMAT(10(5X,A1))
      N=MIN0(N,10)
      WRITE(2,100) (A(I),I=1,N)
      RETURN
      END
```

```
      FUNCTION ASPEC(VAL)
      TEMP=VAL
      IF(VAL.GT.1.0) TEMP=ALOG10(VAL)
      IF(VAL.LT.(-1.0)) TEMP=-ALOG10(-VAL)
      ASPEC=TEMP
      RETURN
      END
```

```
      PROGRAM ACDYN
C
C     ACDYN.92 --- MYOPIC, MULTIPLE (TWO) MISSILES
C
C     USES LF=1.0, DA=0.0 FOR CONTROLS FOR FIRST STEP,
C        THEN MANEUVERS TO MAXIMIZE ACCELERATION NORMAL
C        TO THE "GUIDANCE PLANE" FOR SINGLE MISSILE CRITERION
C
C     FOR COMBINATION, CHECK ALL POSSIBLE ROLLS (+/- 60 DEG)
C        FOR LAMBDA OF 0 AND 1 ONLY TO FIND THE MAX(UC) OF
C        THE MIN(LAMBDA) OF :
C
C         |LAMBDA*ACCEL(DOT)G1|+|(1-LAMBDA)*ACCEL(DOT)G2|
C
C        WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
C        TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
C
C     DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
C        FUNCTION OF LAMBDA; I.E.
C
C              TA1(LAMBDA) = DMIS1(FINAL)
C              TA2(LAMBDA) = DMIS2(FINAL)
C
C     ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
C
C     WRITES TO THE TERMINAL, THEN PRINTS
C
C     RK11 = RK12 = 4.5
C
C     STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
C
C     PLOTTED VARIABLES ARE ---
C
C          PL(,1) = USTAR (DEG)
C          PL(,2) = PERFSTAR
C          PL(,3) = USTAR2 (DEG)
C          PL(,4) = PERF2
C          PL(,5) = GX   )
C          PL(,6) = GY   ) COMBINED USING LAMBDA
C          PL(,7) = GZ   )
C          PL(,8) = LF1 (COMMANDED)
C          PL(,9) = SPECIFIC ENERGY (A/C)
C          PL(,10) = SP. EN. (MISSILE 1)
C          PL(,11) = ALPHA (A/C)
C          PL(,12) = GAMMA (A/C)
C          PL(,13) = SIGMA (A/C)
C          PL(,14) = NORM1(G) = (GX1**2+GY1**2+GZ1**2)**0.5
C          PL(,15) = Z (A/C)
C          PL(,16) = AIRSPEED (A/C)
C          PL(,17) = SY1 (L.O.S. PITCH)
C          PL(,18) = THETA1 (L.O.S. YAW)
C          PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
C          PL(,20) = DRAG (A/C)
C          PL(,21) = Z (MISSILE 1)
C          PL(,22) = X (A/C)
C          PL(,23) = X (MISSILE 1)
C          PL(,24) = Y (A/C)
C          PL(,25) = Y (MISSILE 1)
C          PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2
C          PL(,27) = SYDT1
```

```
C           PL(,28) = THEDT1
C           PL(,29) = NORM1(LOSDOT) = (SYDT1**2+THEDT1**2)**0.5
C           PL(,30) = GX1    )
C           PL(,31) = GY1    )  MISSILE 1
C           PL(,32) = GZ1    )
C           PL(,33) = NORM2
C           PL(,34) = GX2    )
C           PL(,35) = GY2    )  MISSILE 2
C           PL(,36) = GZ2    )
C           PL(,37) = NORM2(LOS-DOT)
C           PL(,38) = LF2C
C           PL(,39) = SP. EN. (MISSILE 2)
C           PL(,40) = SY2 (LOS PITCH)
C           PL(,41) = THETA2 (LOS YAW)
C           PL(,42) = X (MISSILE 2)
C           PL(,43) = Y (MISSILE 2)
C           PL(,44) = Z (MISSILE 2)
C           PL(,45) = SYDT2
C           PL(,46) = THEDT2
C           PL(,47) = LAMDA1 * 100.
C           PL(,48) = GSY1
C           PL(,49) = GTH1
C           PL(,50) = GSY2
C           PL(,51) = GTH2
C           PL(,52) = GSY )
C           PL(,53) = GTH )  COMBINED USING LAMBDA
C
      REAL*8 STRING,SDATE
      INTEGER CUTIME
      LOGICAL PRTED,LONCE
      COMMON /FARM7/ LONCE
      COMMON /FARM9/ ISEC,DMIS1,DMIS2,PRTED
C
      ICUM=CUTIME(0)
      CALL TIMEOD(STRING)
      CALL DATE(SDATE)
      WRITE(6,66) SDATE, STRING
   66 FORMAT(13X,40('*'),' ACDYN.92 ',40('*'),/,
     1   51X,A8,2X,A8,/,1H1)
C
      LONCE=.FALSE.
  999 CALL INIT
      CALL VALUE
      CALL PLOLT(2)
      CALL PLOLT(6)
C
      WRITE(6,100)
  100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 '
     1   'FOR NO INIT PRINT')
      CALL BELL(1)
      READ(1,110) ISEC
  110 FORMAT(I1)
      IF(ISEC.NE.0) GO TO 999
      STOP
      END
```

```
      SUBROUTINE VALUE
      INTEGER LOW(2),HIGH(2)
      REAL MO, LF, LO, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
     1   NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
     2   NORMG
      DIMENSION ACDM(100,2)
      LOGICAL PRTED, FLAG1, FLAG2
      INTEGER CUTINE
      COMMON /AMP/ TO,WCSO,ALFAO,CLO,DO,LO,G,MO,C01,C02,C03,
     1  RHO,BETA,CLAFO,SO,D01,D02,M1,CLAF1,S1,D11,RLMAX,
     2  D12,THRESH,RK11,RK12,G1S1,ALFA1,CL1,D1,L1,ALREAS
     3  ,P1,TAL,TAFTS,TFLAG,W2S2,ALFA2,CL2,D2,L2
      COMMON /FARM1/ JJ,KSTEP
      COMMON /FARM2/ VTHO,VTH1,DTH,TSTEP,ICPUTM
      COMMON /FARM3/ NO,IO
      COMMON /FARM4/ STEP
      COMMON /FARM5/ MANUVR
      COMMON /FARM9/ ISEC,DMIS1,DMIS2,PRTED
      COMMON /ARRAY1/ XO(18),XOIN(18)
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
      COMMON /FLVR/ PL(100,55),PMAX,PMIN,LMAX,LMIN,SFMAX,
     1   SFMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,AMMAX,LOSMAX
      COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2
      COMMON /SINCOS/ TABSIN(360),TABCOS(360)
C
      EQUIVALENCE  (GAMA,XO(5)),(SIGMA,XO(6))
      EQUIVALENCE  (VO,XO(4)),(V1,XO(10)),(V2,XO(16))
C
      DATA TESIN /1.0/, IEXIT/'X'/, IQLIK/'Q'/
C
      COSGAM=COS(GAMA)
      SINGAM=SIN(GAMA)
      COSSIG=COS(SIGMA)
      SINSIG=SIN(SIGMA)
C
      TA=0.0
      JJ=0
      DUM1=1.0E10
      DUM2=1.0E10
      FLAG1=.FALSE.
      FLAG2=.FALSE.
      LAMDA1=0.5
C
C     FLAG(I) INDICATES THAT MISSILE I ALREADY GOT AWAY
C
  100 CONTINUE
      IF(FLAG1) LAMDA1=0.0
      IF(FLAG2) LAMDA1=1.0
      LAMDA2=1.0-LAMDA1
      IVAL=50
      TPRINT=TA
      VOX=VO*COSGAM*COSSIG
      VOY=VO*COSGAM*SINSIG
      VOZ=VO*SINGAM
      V1X=V1*COS(XO(11))*COS(XO(12))
      V1Y=V1*COS(XO(11))*SIN(XO(12))
      V1Z=V1*SIN(XO(11))
      V2X=V2*COS(XO(17))*COS(XO(18))
      V2Y=V2*COS(XO(17))*SIN(XO(18))
```

```
      V2Z=V2*SIN(XC(17))
      VRELX1=VCX-V1X
      VRELY1=VCY-V1Y
      VRELZ1=VCZ-V1Z
      VRELT1=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
      VRELX2=VCX-V2X
      VRELY2=VCY-V2Y
      VRELZ2=VCZ-V2Z
      VRELT2=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
      DELX1=XO(1)-XC(7)
      DELY1=XO(2)-XC(8)
      DELZ1=XO(3)-XC(9)
      DELX2=XO(1)-XO(13)
      DELY2=XO(2)-XO(14)
      DELZ2=XO(3)-XO(15)
C
      DUM1=AMIN1(DUM1,DMIS1)
      DUM2=AMIN1(DUM2,DMIS2)
      DMIS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
      DMIS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
      DSV1=0.5*DMIS1/VRELT1
      DSV2=0.5*DMIS2/VRELT2
      STEP=AMIN1(TSTEP,DSV1,DSV2)
      DMIS=AMIN1(DMIS1,DMIS2)
      IF(JJ.GT.0) MANUVR=1
      LF=1.0
      ULAST=UO
      UO=0.0
      IF(MANUVR.NE.1) GO TO 300
C
      UO=ULAST
      LF=MAXLF
      ISEC=ISEC+1
      GX1 = VRELY1*DELZ1 - VRELZ1*DELY1
      GY1 = VRELZ1*DELX1 - VRELX1*DELZ1
      GZ1 = VRELX1*DELY1 - VRELY1*DELX1
      NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
      GX2 = VRELY2*DELZ2 - VRELZ2*DELY2
      GY2 = VRELZ2*DELX2 - VRELX2*DELZ2
      GZ2 = VRELX2*DELY2 - VRELY2*DELX2
      NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
      IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 200
C
C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
C
      LF=ACLF(ISEC-1)
      GO TO 290
C
  200 CONTINUE
      GX1=GX1/NORM1
      GY1=GY1/NORM1
      GZ1=GZ1/NORM1
      GX2=GX2/NORM2
      GY2=GY2/NORM2
      GZ2=GZ2/NORM2
C
      COEFA1=GZ1*COSGAM-GY1*SINGAM*SINSIG-GX1*SINGAM*COSSIG
      COEFA1=COEFA1*(LO+TO*SIN(ALFAO))
      COEFB1=GY1*COSGAM-GX1*SINGAM
      COEFB1=COEFB1*(LO+TO*SIN(ALFAO))
```

```
      COEFC1=((TO*COS(ALFAO)-DO)*(GZ1*SINGAM+GY1*COSGAM*SINSIG
     1   +GX1*(COSGAM*COSSIG)) - GZ1*MO*G
C
      COEFA2=GZ2*COSGAM-GY2*SINGAM*SINSIG-GX2*SINGAM*COSSIG
      COEFA2=COEFA2*(LO+TO*SIN(ALFAO))
      COEFB2=GY2*COSGAM-GX2*SINGAM
      COEFB2=COEFB2*(LO+TO*SIN(ALFAO))
      COEFC2=((TO*COS(ALFAO)-DO)*(GZ2*SINGAM+GY2*COSGAM*SINSIG
     2   +GX2*(COSGAM*COSSIG)) - GZ2*MO*G
C
C     CHECK EVERY ANGLE (BY DEGREE) WITHIN +/- RLMAX
C
      NBLK=1
      PRFMAX=0.0
      IUO=(UO*180./PI)+180.
      IF(IUO.GT.360) IUO=IUO-360
      IF(IUO.LT.1) IUO=IUO+360
      IROLL=RLMAX*STEP
      LOW(1)=IUO-IROLL
      IF(LOW(1).GE.1) GO TO 210
      NBLK=2
      LOW(1)=1
      LOW(2)=360+IUO-IROLL
      HIGH(2)=360
  210 CONTINUE
      HIGH(1)=IUO+IROLL
      IF(HIGH(1).LE.360) GO TO 220
      NBLK=2
      HIGH(1)=IUO+IROLL-360
      HIGH(2)=360
      LOW(2)=LOW(1)
      LOW(1)=1
  220 CONTINUE
      DO 230 IJK=1,NBLK
      ISTRT=LOW(IJK)
      IEND=HIGH(IJK)
      DO 230 IUO=ISTRT,IEND
      PERF1=COEFA1*TABCOS(IUO)+COEFB1*TABSIN(IUO)+COEFC1
      PERF2=COEFA2*TABCOS(IUO)+COEFB2*TABSIN(IUO)+COEFC2
      ABS1=ABS(PERF1)
      ABS2=ABS(PERF2)
      IF(FLAG1) ABS1=ABS2*2.0
      IF(FLAG2) ABS2=ABS1*2.0
      PRFMIN=AMIN1(ABS1,ABS2)
      IF(PRFMIN.LE.PRFMAX) GO TO 230
      IFLAG=IVAL
      IMAX=IUO
      PRFMAX=PRFMIN
      IF(ABS1.LE.ABS2) GO TO 230
      IFLAG=-IVAL
  230 CONTINUE
      UO=FLOAT(IMAX-180)*PI/180.
C
C     NOTE --- IFLAG IS POSITIVE FOR LAMBDA = 1, AND NEGATIVE
C        FOR LAMBDA = 0
C
      PL(ISEC,2)=PERF1
      PL(ISEC,4)=PERF2
      PL(ISEC,26)=IFLAG
      PMAX=AMAX1(PMAX,PERF1,PERF2)
```

```
      PMIN=AMIN1(PMIN,PERF1,PERF2)
C
C     CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
C
      ACCX=((TO*COS(ALFAO)-DO)*(COSGAM*COSSIG))-
     1   ((LO+TO*SIN(ALFAO))*(SIN(UO)*SINGAM+COS(UO)*
     2   SINGAM*COSSIG))
      ACCY=((TO*COS(ALFAO)-DO)*(COSGAM*SINSIG))+
     1   ((LO+TO*SIN(ALFAO))*(SIN(UO)*COSGAM-COS(UO)*
     2   SINGAM*SINSIG))
      ACCZ=((TO*COS(ALFAO)-DO)*SINGAM)-(MO*G)+
     1   ((LO+TO*SIN(ALFAO))*COSGAM*COS(UO))
      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
C
C     STORE PLOTTED VARIABLES
C
      PL(ISEC,14)=NORM1
      PL(ISEC,19)=ANORM/MO
      PL(ISEC,30)=GX1
      PL(ISEC,31)=GY1
      PL(ISEC,32)=GZ1
      PL(ISEC,33)=NORM2
      PL(ISEC,34)=GX2
      PL(ISEC,35)=GY2
      PL(ISEC,36)=GZ2
      NORMAX=AMAX1(NORMAX,NORM1,NORM2)
      ANMAX=AMAX1(ANMAX,ANORM)
C
  290 CONTINUE
      IF(PRTED) GO TO 300
      PRTED=.TRUE.
      CC1=CC1+TAFTB*TFLAG
      TCHG=TPRINT
      WRITE(2,310) TPRINT
      WRITE(6,310) TPRINT
  310 FORMAT(2X,"START MANEUVER AT T = ",F6.2)
      CALL BELL(3)
C
      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
      ALFMAX=ALFAO
      ALFMIN=ALFAO
      ZMAX=AMIN1(XO(3),XO(9),XO(15))
      ZMIN=AMAX1(XO(3),XO(9),XO(15))
      XMAX=AMAX1(XO(1),XO(7),XO(13))
      XMIN=AMIN1(XO(1),XO(7),XO(13))
      YMAX=AMAX1(XO(2),XO(8),XO(14))
      YMIN=AMIN1(XO(2),XO(8),XO(14))
      VMAX=VO
      VMIN=VO
      DMAX=DO
      DMIN=DO
      LOSMAX=0.0
C
  300 CONTINUE
      ACSA(ISEC)=UO*180./PI
      ACLF(ISEC)=LF
      ACTI(ISEC)=TPRINT
      ACDM(ISEC,1)=DMIS1
      ACDM(ISEC,2)=DMIS2
```

```
C
      IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
      WRITE (2,400) TPRINT, DMIS1, DMIS2
      WRITE (6,400) TPRINT, DMIS1, DMIS2
  400 FORMAT (5X,'TIME = ',F10.3,2X,'DSEP1 = ',G12.3,2X,
     1     'DSEP2 = ',G12.3)
C
  477 CONTINUE
      IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
      IF(DMIS.LT.DTH) GO TO 520
      IF(TA.LT.3.0) GO TO 480
      FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
      FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
      IF(FLAG1.AND.FLAG2) GO TO 540
  480 CONTINUE
      CALL INTEOX
      TA=TA+STEP
      JJ=JJ+1
      IF(ISEC.GE.10) GO TO 530
      GO TO 100
C
  510 CONTINUE
      WRITE (2,515) TPRINT
      WRITE (6,515) TPRINT
  515 FORMAT(1X,'**   A/C OR MISSILE VEL. IS TOO LOW ',
     1 'AT TIME: ',F10.3,'   **',/)
      GO TO 600
C
C     HIT OCCURRED, PRINT OUT
  520 CONTINUE
      WRITE(2,525) TPRINT,DUM1,DUM2
      WRITE(6,525) TPRINT,DUM1,DUM2
  525 FORMAT(1X,'*****  HIT  AT TIME = ',F10.3,
     1    '  *****',/,5X,'BEST DSEPS WERE 1:',
     2   G15.6,', & 2:',G15.6,/)
      GO TO 600
C
  530 CONTINUE
      WRITE(2,535) TPRINT
      WRITE(6,535) TPRINT
  535 FORMAT(5X,'TIME LIMIT AT T = ',F6.2)
      GO TO 600
C
C     CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
  540 CONTINUE
      WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
      WRITE(6,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
  544 FORMAT(3X,'*** CLOSURE RATE NEGATIVE AT TIME = ',F10.3,
     1    ' ***',/,5X,'TA1 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/,
     2    5X,'TA2 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/)
      GO TO 600
C
  600 CONTINUE
      CALL BELL(1)
C
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      N10=6
      DO 620 J=1,N10
      JJJ=J+6
```

```
      KK=J+12
      TEMP=X0(J)
      TEMP1=X0(JJJ)
      TEMP2=X0(KK)
      IF(J.LT.5) GO TO 605
      TEMP=TEMP*180./PI
      TEMP1=TEMP1*180./PI
      TEMP2=TEMP2*180./PI
  605 CONTINUE
      WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
      WRITE(6,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
  610 FORMAT(2X,'X0(',I1,'): ',G13.4,4X,
     1   'X0(',I2,'): ',G13.4,4X,
     2   'X0(',I2,'): ',G13.4)
  620 CONTINUE
C
      WRITE (2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
      WRITE (6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
  625 FORMAT(/,10X,'DELX1: ',G12.3,2X,'DELY1: ',G12.3,/,
     1  10X,'DELZ1: ',G12.3,2X,'DMIS1: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      WRITE (2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
      WRITE (6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
  630 FORMAT(/,10X,'DELX2: ',G12.3,2X,'DELY2: ',G12.3,/,
     1  10X,'DELZ2: ',G12.3,2X,'DMIS2: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      CALL BELL(1)
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
  635 FORMAT(A1)
C
      WRITE(6,631)
  631 FORMAT(1H1)
      DO 650 I=1,ISEC
      IF(MOD(I,20).EQ.0) CALL BELL(1)
      IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
      WRITE(6,636) ACTI(I),ACBA(I),PL(I,8),PL(I,38)
  636 FORMAT(1X,'TIME ',F5.2,3X,'ACBA ',F6.2,3X,'LF1C ',F8.1,
     1   3X,'LF2C ',F8.1)
  650 CONTINUE
  660 CONTINUE
      DO 670 I=1,ISEC
      WRITE(6,637) ACTI(I),ACLF(I),ACPA(I),PL(I,8),PL(I,38),
     1   PL(I,19),PL(I,14),PL(I,11),PL(I,20),
     2   (ACUM(I,J),J=1,2),PL(I,47)
  637 FORMAT(1X,'TIME = ',0PF6.2,3X,'ACLF = ',F5.1,3X,
     1 'ACBA(DEG) = ',F8.2,3X,'LF1C = ',F8.1,3X,
     2   'LF2C = ',F8.2,3X,'NORM(A) = ',G11.2,/,20X,
     3   'NORM1(G) = ',G11.2,3X,'ALPHA = ',F8.2,3X,
     4   'DRAG = ',G11.2,/,20X,'DMIS1 = TA(1) = ',
     5   F9.2,5X,'DMIS2 = TA(2) = ',F9.2,5X,
     6   'LAMBDA1 = ',-2PF5.2,/)
  670 CONTINUE
C
      IDUM=CUTIME(0)-ICPUTM
      WRITE(2,639) IDUM
      WRITE(6,639) IDUM
```

```
  639 FORMAT(4X,'CPU TIME (SECONDS) = ',I5,0PF5.0)
C
      RETURN
      END
```

```
      PROGRAM ACDYN
C
C     ACDYN.93 --- MYOPIC, MULTIPLE (TWO) MISSILES
C
C     USES LF=1.0, LA=0.0 FOR CONTROLS FOR FIRST STEP,
C        THEN MANEUVERS TO MAXIMIZE ACCELERATION IN THE
C        "GUIDANCE PLANE" ANTI-PROPORTIONALLY TO THE
C        MISSILE'S GUIDANCE
C
C     FOR COMBINATION, USE DO LOOP OF LAMBDA'S, I.E.
C        LAMBDA = 0.0 THRU 1.0 BY 0.05, AT EACH STEP,
C        TO FIND THE MIN(LAMBDA) OF THE MAX(UO) OF :
C
C          |(LAMBDA*ACCEL(DOT)G1)+((1-LAMBDA)*ACCEL(DOT)G2)|
C
C        WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
C        TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
C
C     DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
C        FUNCTION OF LAMBDA; I.E.
C
C              TA1(LAMBDA) = DMIS1(FINAL)
C              TA2(LAMBDA) = DMIS2(FINAL)
C
C     ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
C
C     WRITES TO THE TERMINAL, THEN PRINTS
C
C     RK11 = RK12 = 4.5
C
C     STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
C
C     PLOTTED VARIABLES ARE ---
C
C          PL(,1) = USTAR (DEG)
C          PL(,2) = PERFSTAR
C          PL(,3) = USTAR2 (DEG)
C          PL(,4) = PERF2
C          PL(,5) = GX  )
C          PL(,6) = GY  ) COMBINED USING LAMBDA
C          PL(,7) = GZ  )
C          PL(,8) = LF1 (COMMANDED)
C          PL(,9) = SPECIFIC ENERGY (A/C)
C          PL(,10) = SP. EN. (MISSILE 1)
C          PL(,11) = ALPHA (A/C)
C          PL(,12) = GAMMA (A/C)
C          PL(,13) = SIGMA (A/C)
C          PL(,14) = NORM1(G) = (GX1**2+GY1**2+GZ1**2)**0.5
C          PL(,15) = Z (A/C)
C          PL(,16) = AIRSPEED (A/C)
C          PL(,17) = SY1 (L.O.S. PITCH)
C          PL(,18) = THETA1 (L.O.S. YAW)
C          PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
C          PL(,20) = DRAG (A/C)
C          PL(,21) = Z (MISSILE 1)
C          PL(.22) = X (A/C)
C          PL(,23) = X (MISSILE 1)
C          PL(,24) = Y (A/C)
C          PL(,25) = Y (MISSILE 1)
C          PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2
```

```
C             PL(,27) = SYDT1
C             PL(,28) = THEDT1
C             PL(,29) = NORM1(LOSDOT) = (SYDT1**2+THEDT1**2)**0.5
C             PL(,30) = GX1   )
C             PL(,31) = GY1   )  MISSILE 1
C             PL(,32) = GZ1   )
C             PL(,33) = NORM2
C             PL(,34) = GX2   )
C             PL(,35) = GY2   )  MISSILE 2
C             PL(,36) = GZ2   )
C             PL(,37) = NORM2(LOS-DOT)
C             PL(,38) = LF2C
C             PL(,39) = SP. EN. (MISSILE 2)
C             PL(,40) = SY2 (LOS PITCH)
C             PL(,41) = THETA2 (LOS YAW)
C             PL(,42) = X (MISSILE 2)
C             PL(,43) = Y (MISSILE 2)
C             PL(,44) = Z (MISSILE 2)
C             PL(,45) = SYDT2
C             PL(,46) = THEDT2
C             PL(,47) = LAMDA1 * 100.
C             PL(,48) = GSY1
C             PL(,49) = GTH1
C             PL(,50) = GSY2
C             PL(,51) = GTH2
C             PL(,52) = GSY )
C             PL(,53) = GTH )  COMBINED USING LAMBDA
C
      REAL*8 STRING,SDATE
      INTEGER CUTIME
      LOGICAL PRTED,LONCE
      COMMON /PARM7/ LONCE
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
C
      IDUM=CUTIME(0)
      CALL TIMEOD(STRING)
      CALL DATE(SDATE)
      WRITE(5,66) SDATE, STRING
   66 FORMAT(15X,40('*'),' ACDYN.93 ',40('*'),/,
     1   51X,A8,2X,A8,/,1H1)
C
      LONCE=.FALSE.
  999 CALL INIT
      CALL VALUE
      CALL PLOUT(2)
      CALL PLOUT(6)
C
      WRITE(2,100)
  100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 '
     1   'FOR NO INIT PRINT')
      CALL BELL(1)
      READ(1,110) ISEC
  110 FORMAT(I1)
      IF(ISEC.NE.0) GO TO 999
      STOP
      END
```

```
      SUBROUTINE VALUE
      REAL MO, LF, LO, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
     1   NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
     2   NORMG
      DIMENSION ACUM(100,2)
      LOGICAL PRTED, FLAG1, FLAG2
      INTEGER CUTIME
      COMMON /AMP/ TO,MCSO,ALFAO,CLO,DO,LO,G,MO,CO1,CO2,CO3,
     1  RHO,BETA,CLAFO,SO,DO1,DO2,M1,CLAF1,S1,D11,RLMAX,
     2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALMEAS
     3  ,PI,TAU,TAFTB,TFLAG,Q2S2,ALFA2,CL2,D2,L2
      COMMON /PARM1/ JJ,KSTEP
      COMMON /PARM2/ VTHO,VTH1,DTH,TSTEP,ICPUTM
      COMMON /PARM3/ NO,IO
      COMMON /PARM4/ STEP
      COMMON /PARM5/ MANUVR
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
      COMMON /ARRAY1/ XO(13),XOIN(18)
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,UO,MAXLF
      COMMON /PLVR/ PL(100,53),PMAX,PMIN,LMAX,LMIN,SPMAX,
     1   SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
      COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2,
     1   SYDT1,THEDT1,SYDT2,THEDT2
C
      EQUIVALENCE (GAMA,XO(5)),(SIGMA,XO(6))
      EQUIVALENCE (VO,XO(4)),(V1,XO(10)),(V2,XO(16))
C
      DATA TESTN /1.0E-06/, IEXIT/'X'/, IQUIK/'Q'/
C
      PERF(U)=COEFA*COS(U)+COEFB*SIN(U)+COEFC
C
      COSGAM=COS(GAMA)
      SINGAM=SIN(GAMA)
      COSSIG=COS(SIGMA)
      SINSIG=SIN(SIGMA)
      COSSY1=COS(PSY1)
      COSSY2=COS(PSY2)
      COSTH1=COS(THETA1)
      COSTH2=COS(THETA2)
      SINSY1=SIN(PSY1)
      SINSY2=SIN(PSY2)
      SINTH1=SIN(THETA1)
      SINTH2=SIN(THETA2)
C
      TA=0.0
      JJ=0
      DUM1=1.0E10
      DUM2=1.0E10
      FLAG1=.FALSE.
      FLAG2=.FALSE.
      LAMDA1=0.5
C
C     FLAG(I) INDICATES THAT MISSILE I ALREADY GOT AWAY
C
  100 CONTINUE
      IF(FLAG1) LAMDA1=0.0
      IF(FLAG2) LAMDA1=1.0
      LAMDA2=1.0-LAMDA1
      IVAL=50
```

```
      TPRINT=TA
      VOX=V0*COSGAM*COSSIG
      VOY=V0*COSGAM*SINSIG
      VOZ=V0*SINGAM
      V1X=V1*COS(XO(11))*COS(XO(12))
      V1Y=V1*COS(XO(11))*SIN(XO(12))
      V1Z=V1*SIN(XO(11))
      V2X=V2*COS(XO(17))*COS(XO(13))
      V2Y=V2*COS(XO(17))*SIN(XO(13))
      V2Z=V2*SIN(XO(17))
      VRELX1=VOX-V1X
      VRELY1=VOY-V1Y
      VRELZ1=VOZ-V1Z
      VRELT1=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
      VRELX2=VOX-V2X
      VRELY2=VOY-V2Y
      VRELZ2=VOZ-V2Z
      VRELT2=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
      DELX1=XO(1)-XO(7)
      DELY1=XO(2)-XO(8)
      DELZ1=XO(3)-XO(9)
      DELX2=XO(1)-XO(13)
      DELY2=XO(2)-XO(14)
      DELZ2=XO(3)-XO(15)
C
      DUM1=AMIN1(DUM1,DMIS1)
      DUM2=AMIN1(DUM2,DMIS2)
      DMIS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
      DMIS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
      DSV1=0.5*DMIS1/VRELT1
      DSV2=0.5*DMIS2/VRELT2
      STEP=AMIN1(TSTEP,DSV1,DSV2)
      DMIS=AMIN1(DMIS1,DMIS2)
      IF(J).GT.0) MANUVR=1
      LF=1.0
      ULAST=U0
      U0=0.0
      IF(MANUVR.NE.1) GO TO 300
C
      U0=ULAST
      LF=MAXLF
      ISEC=ISEC+1
      GX1 = -(SYDT1*SINSY1*COSTH1 + THEDT1*COSSY1*SINTH1)
      GY1 = THEDT1*COSSY1*COSTH1 - SYDT1*SINSY1*SINTH1
      GZ1 = SYDT1* COSSY1
      NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
      GX2 = -(SYDT2*SINSY2*COSTH2 + THEDT2*COSSY2*SINTH2)
      GY2 = THEDT2*COSSY2*COSTH2 - SYDT2*SINSY2*SINTH2
      GZ2 = SYDT2* COSSY2
      NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
      IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 200
C
C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
C
      LF=ACLF(ISEC-1)
      GO TO 290
C
  200 CONTINUE
      GX1=GX1/NORM1
      GY1=GY1/NORM1
```

```
      GZ1=GZ1/NORM1
      GX2=GX2/NORM2
      GY2=GY2/NORM2
      GZ2=GZ2/NORM2
      PERFMN=1.0E20
C
      DO 270 LAMB=1,21
      LAMDA1=FLOAT(LAMB-1)/20.
      LAMDA2=1.0-LAMDA1
      IF(FLAG1.AND.(LAMB.GT.1)) GO TO 280
      IF(FLAG2.AND.(LAMB.LT.21)) GO TO 270
C
      GX=LAMDA1*GX1+LAMDA2*GX2
      GY=LAMDA1*GY1+LAMDA2*GY2
      GZ=LAMDA1*GZ1+LAMDA2*GZ2
C
      COEFA=GZ*COSGAM-GY*SINGAM*SINSIG-GX*SINGAM*COSSIG
      COEFA=COEFA*(L0+T0*SIN(ALFA0))
      COEFB=GY*COSGAM-GX*SINGAM
      COEFB=COEFB*(L0+T0*SIN(ALFA0))
      COEFC=((T0*COS(ALFA0)-D0))*(GZ*SINGAM+GY*COSGAM*SINSIG
     1   +GX*COSGAM*COSSIG)) - GZ*M0*G
C
      USTAR=ATAN2(COEFB,COEFA)
      USTAR2=USTAR+PI
      IF(USTAR.GT.0.0) USTAR2=USTAR-PI
  220 CONTINUE
      PERFST=PERF(USTAR)
      PERF2=PERF(USTAR2)
      ABST=ABS(PERFST)
      ABS2=ABS(PERF2)
C
C     USE "BEST" U AND RATE LIMIT TO RLMAX
C
      IFLAG=IVAL
      UMAX=USTAR
      PERMIN=ABST
      IF(ABST.GE.ABS2) GO TO 230
      UMAX=USTAR2
      IFLAG=-IVAL
      PERMIN=ABS2
  230 CONTINUE
      ROLL=(UMAX-U0)*180./PI
      IF(ROLL.GT.180.) ROLL=ROLL-360.
      IF(ROLL.LT.(-180.)) ROLL=ROLL+360.
      AROLL=ABS(ROLL)
C
C     WE NOW RATE-LIMIT WHICHEVER ANGLE WE GET TO RLMAX
C          DEG PER SEC...
C
      IF(AROLL.LE.(RLMAX*STEP)) GO TO 250
      ROLL=ROLL*RLMAX*STEP/AROLL
      IF(AROLL.LT.180.) GO TO 250
C
C     ROLL IS ESSENTIALLY A COMPLETE FLIP --- USE PLUS/MINUS
C         RLMAX INTO PERF FUNCTION TO CHECK FOR BEST WAY
C         TO MAKE FLIPS
C
      IVAL=75
      USTAR=U0+(RLMAX*STEP*PI/180.)
```

```
      IF(USTAR.GT.PI) USTAR=USTAR-(2.0*PI)
      USTAR2=U0-(RLMAX*STEP*PI/180.)
      IF(USTAR2.LE.(-PI)) USTAR2=USTAR2+(2.0*PI)
      GO TO 220
C
  250 CONTINUE
      IF(PLTMIN.GE.PERFM) GO TO 270
      PERFM=PERMIN
      JMIN=LAMU
      RLMIN=ROLL
      PL(ISEC,1)=USTAR*180./PI
      PL(ISEC,2)=PERFST
      PL(ISEC,3)=USTAR2*180./PI
      PL(ISEC,4)=PERF2
      PL(ISEC,5)=GX
      PL(ISEC,6)=GY
      PL(ISEC,7)=GZ
      PL(ISEC,26)=IFLAG
      PL(ISEC,47)=5*(JMIN-1)
      PMAX=AMAX1(PMAX,PERFST,PERF2)
      PMIN=AMIN1(PMIN,PERFST,PERF2)
  270 CONTINUE
  280 CONTINUE
C
      ROLL=RLMIN
      U0=U0+(ROLL*PI/180.)
      IF(U0.GT.PI) U0=U0-(2.0*PI)
      IF(U0.LE.(-PI)) U0=U0+(2.0*PI)
C
C     CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
C
      ACCX=((T0*COS(ALFA0)-D0)*(COSGAM*COSSIG))-
     1   ((L0+T0*SIN(ALFA0))*(SIN(U0)*SINGAM+COS(U0)*
     2   SINGAM*COSSIG))
      ACCY=((T0*COS(ALFA0)-D0)*(COSGAM*SINSIG))+
     1   ((L0+T0*SIN(ALFA0))*(SIN(U0)*COSGAM-COS(U0)*
     2   SINGAM*SINSIG))
      ACCZ=((T0*COS(ALFA0)-D0)*SINGAM)-(M0*G)+
     1   ((L0+T0*SIN(ALFA0))*COSGAM*COS(U0))
      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
C
C     STORE PLOTTED VARIABLES
C
      PL(ISEC,14)=NORM1
      PL(ISEC,19)=ANORM/M0
      PL(ISEC,30)=GX1
      PL(ISEC,31)=GY1
      PL(ISEC,32)=GZ1
      PL(ISEC,33)=NORM2
      PL(ISEC,34)=GX2
      PL(ISEC,35)=GY2
      PL(ISEC,36)=GZ2
      NORMAX=AMAX1(NORMAX,NORM1,NORM2)
      ANMAX=AMAX1(ANMAX,ANORM)
C
  290 CONTINUE
      IF(PRTED) GO TO 300
      PRTED=.TRUE.
      C01=C01+TAFTG*TFLAG
      TCHG =TPRINT
```

```
      WRITE(2,310) TPRINT
      WRITE(6,310) TPRINT
  310 FORMAT(2X,'START MANEUVER AT T = ',F6.2)
      CALL BELL(3)
C
      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
      ALFMAX=ALFAO
      ALFMIN=ALFAO
      ZMAX=AMIN1(XO(3),XO(9),XO(15))
      ZMIN=AMAX1(XO(3),XO(9),XO(15))
      XMAX=AMAX1(XO(1),XO(7),XO(13))
      XMIN=AMIN1(XO(1),XO(7),XO(13))
      YMAX=AMAX1(XO(2),XO(8),XO(14))
      YMIN=AMIN1(XO(2),XO(8),XO(14))
      VMAX=V0
      VMIN=V0
      DMAX=D0
      DMIN=D0
      LOSMAX=0.0
C
  300 CONTINUE
      ACBA(ISEC)=U0*180./PI
      ACLF(ISEC)=LF
      ACTI(ISEC)=TPRINT
      ACDM(ISEC,1)=DMIS1
      ACDM(ISEC,2)=DMIS2
C
      IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
      WRITE (2,400) TPRINT, DMIS1, DMIS2
      WRITE (6,400) TPRINT, DMIS1, DMIS2
  400 FORMAT (5X,'TIME = ',F10.3,2X,'DSEP1 = ',G12.3,2X,
     1     'DSEP2 = ',G12.3)
C
  477 CONTINUE
      IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
      IF(DMIS.LT.DTH) GO TO 520
      IF(TA.LT.3.0) GO TO 480
      FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
      FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
      IF(FLAG1.AND.FLAG2) GO TO 540
  480 CONTINUE
      CALL INTBOX
      TA=TA+STEP
      JJ=JJ+1
      IF(ISEC.GE.IO) GO TO 530
      GO TO 100
C
  510 CONTINUE
      WRITE (2,515) TPRINT
      WRITE (6,515) TPRINT
  515 FORMAT(1X,'**   A/C OR MISSILE VEL. IS TOO LOW ',
     1 'AT TIME: ',F10.3,'    **',/)
      GO TO 600
C
C     HIT OCCURRED, PRINT OUT
  520 CONTINUE
      WRITE(2,525) TPRINT,DUM1,DUM2
      WRITE(6,525) TPRINT,DUM1,DUM2
  525 FORMAT(1X,'*****  HIT  AT TIME = ',F10.3,
```

```
     1    '  *****',/,5X,'BEST DSEPS WERE 1:',
     2    G15.6,', & 2:',G15.6,/)
      GO TO 600
C
  530 CONTINUE
      WRITE(2,535) TPRINT
      WRITE(6,535) TPRINT
  535 FORMAT(5X,'TIME LIMIT AT T = ',F6.2)
      GO TO 600
C
C     CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
  540 CONTINUE
      WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
      WRITE(6,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
  544 FORMAT(3X,'**** CLOSURE RATE NEGATIVE AT TIME = ',F10.3,
     1    '****',/,5X,'TA1 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/,
     2    5X,'TA2 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/)
      GO TO 600
C
  600 CONTINUE
      CALL BELL(1)
C
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      N10=6
      DO 620 J=1,N10
      JJJ=J+6
      KK=J +12
      TEMP=XO(J)
      TEMP1=XO(JJJ)
      TEMP2=XO(KK)
      IF(J.LT.5) GO TO 605
      TEMP=TEMP*180./PI
      TEMP1=TEMP1*180./PI
      TEMP2=TEMP2*180./PI
  605 CONTINUE
      WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
      WRITE(6,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
  610 FORMAT(2X,'XO(',I1,'): ',G13.4,4X,
     1    'XO(',I2,'): ',G13.4,4X,
     2    'XO(',I2,'): ',G13.4)
  620 CONTINUE
C
      WRITE (2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
      WRITE (6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
  625 FORMAT(/,10X,'DELX1: ',G12.3,2X,'DELY1: ',G12.3,/,
     1  10X,'DELZ1: ',G12.3,2X,'DMIS1: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      WRITE (2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
      WRITE (6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
  630 FORMAT(/,10X,'DELX2: ',G12.3,2X,'DELY2: ',G12.3,/,
     1  10X,'DELZ2: ',G12.3,2X,'DMIS2: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      CALL BELL(1)
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
  635 FORMAT(A1)
C
      WRITE(6,631)
```

```
631 FORMAT(1H1)
    DO 650 I=1,ISEC
    IF(MOD(I,20).EQ.0) CALL BELL(1)
    IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
    IF(LOGIC.EQ.IEXIT) RETURN
    IF(LOGIC.EQ.IQUIK) GO TO 660
    WRITE(2,636) ACTI(I),ACBA(I),PL(I,8),PL(I,38)
636 FORMAT(1X,'TIME ',F5.2,3X,'ACBA ',F8.2,3X,'LF1C ',F8.1,
   1   3X,'LF2C ',F8.1)
650 CONTINUE
660 CONTINUE
    DO 670 I=1,ISEC
    WRITE(6,637) ACTI(I),ACLF(I),ACBA(I),PL(I,8),PL(I,38),
   1   PL(I,19),PL(I,14),PL(I,11),PL(I,20),
   2   (ACUM(I,J),J=1,2),PL(I,47)
637 FORMAT(1X,'TIME = ',0PF6.2,3X,'ACLF = ',F5.1,3X,
   1 'ACBA(DEG) = ',F5.2,3X,'LF1C = ',F8.1,3X,
   2   'LF2C = ',F8.2,3X,'NORM(A) = ',G11.2,/,20X,
   3   'NORM1(G) = ',G11.2,3X,'ALPHA = ',F8.2,3X,
   4   'DRAG = ',G11.2,/,20X,'DMIS1 = TA(1) = ',
   5   F5.2,5X,'DMIS2 = TA(2) = ',F9.2,5X,
   6   'LAMBDA1 = ',-2PF5.2,/)
670 CONTINUE
C
    IDUM=CUTIME(0)-ICPUTM
    WRITE(2,639) IDUM
    WRITE(6,639) IDUM
639 FORMAT(4X,'CPU TIME (SECONDS) = ',I5,0PF5.0)
C
    RETURN
    END
```

```
      PROGRAM ACDYN
C
C     ACDYN.94 --- MYOPIC, MULTIPLE (TWO) MISSILES
C
C     USES LF=1.0, FA=0.0 FOR CONTROLS FOR FIRST STEP,
C        THEN MANEUVERS TO MAXIMIZE ACCELERATION IN THE
C        "GUIDANCE PLANE" ANTI-PROPORTIONALLY TO THE
C        MISSILE'S GUIDANCE
C
C     FOR COMBINATION, CHECK ALL POSSIBLE ROLLS (+/- 60 DEG)
C        FOR LAMBDA OF 0 AND 1 ONLY TO FIND THE MAX(U0) OF
C        THE MIN(LAMBDA) OF :
C
C         |(LAMBDA*ACCEL(DOT)G1)|+|(1-LAMBDA)*ACCEL(DOT)G2)|
C
C        WHEN ONE MISSILE HAS MISSED THE AIRCRAFT, BEGIN
C        TO IGNORE IT --- I.E., SET LAMBDA TO 1 OR 0
C
C     DEFINE THREAT ASSESSMENT FOR EACH MISSILE I AS A
C        FUNCTION OF LAMBDA; I.E.
C
C              TA1(LAMBDA) = DMIS1(FINAL)
C              TA2(LAMBDA) = DMIS2(FINAL)
C
C     ALL MANEUVERS ROLL-RATE LIMITED (RLMAX DEGREES)
C
C     WRITES TO THE TERMINAL, THEN PRINTS
C
C     RK11 = RK12 = 4.5
C
C     STORAGE FOR UP TO 100 ITERATIONS AFTER ONSET OF MANEUVER
C
C     PLOTTED VARIABLES ARE ---
C
C          PL(,1) = USTAR (DEG)
C          PL(,2) = PERFSTAR
C          PL(,3) = USTAR2 (DEG)
C          PL(,4) = PERF2
C          PL(,5) = GX   )
C          PL(,6) = GY   ) COMBINED USING LAMBDA
C          PL(,7) = GZ   )
C          PL(,8) = LF1 (COMMANDED)
C          PL(,9) = SPECIFIC ENERGY (A/C)
C          PL(,10) = SP. EN. (MISSILE 1)
C          PL(,11) = ALPHA (A/C)
C          PL(,12) = GAMMA (A/C)
C          PL(,13) = SIGMA (A/C)
C          PL(,14) = NORM1(G) = (GX1**2+GY1**2+GZ1**2)**0.5
C          PL(,15) = Z (A/C)
C          PL(,16) = AIRSPEED (A/C)
C          PL(,17) = SY1 (L.O.S. PITCH)
C          PL(,18) = THETA1 (L.O.S. YAW)
C          PL(,19) = NORM(ACCEL) = (AX**2+AY**2+AZ**2)**0.5
C          PL(,20) = DRAG (A/C)
C          PL(,21) = Z (MISSILE 1)
C          PL(,22) = X (A/C)
C          PL(,23) = X (MISSILE 1)
C          PL(,24) = Y (A/C)
C          PL(,25) = Y (MISSILE 1)
C          PL(,26) = IFLAG = 50, + FOR USTAR, - FOR USTAR2
```

```
C          PL(,27) = SYDT1
C          PL(,28) = THEDT1
C          PL(,29) = NORM1(LOSDOT) = (SYDT1**2+THEDT1**2)**0.5
C          PL(,30) = GX1    )
C          PL(,31) = GY1    )  MISSILE 1
C          PL(,32) = GZ1    )
C          PL(,33) = NORM2
C          PL(,34) = GX2    )
C          PL(,35) = GY2    )  MISSILE 2
C          PL(,36) = GZ2    )
C          PL(,37) = NORM2(LOS-DOT)
C          PL(,38) = LF2C
C          PL(,39) = SP. EN. (MISSILE 2)
C          PL(,40) = SY2 (LOS PITCH)
C          PL(,41) = THETA2 (LOS YAW)
C          PL(,42) = X (MISSILE 2)
C          PL(,43) = Y (MISSILE 2)
C          PL(,44) = Z (MISSILE 2)
C          PL(,45) = SYDT2
C          PL(,46) = THEDT2
C          PL(,47) = LAMDA1 * 100.
C          PL(,48) = GSY1
C          PL(,49) = GTH1
C          PL(,50) = GSY2
C          PL(,51) = GTH2
C          PL(,52) = GSY )
C          PL(,53) = GTH )  COMBINED USING LAMBDA
C
      REAL*8 STRING,SDATE
      INTEGER CUTIME
      LOGICAL PRTED,LONCE
      COMMON /PARM7/ LONCE
      COMMON /PARMS/ ISEC,DMIS1,DMIS2,PRTED
C
      IDUM=CUTIME(0)
      CALL TIMEOD(STRING)
      CALL DATE(SDATE)
      WRITE(6,66) SDATE, STRING
   66 FORMAT(15X,40('*'),' ACDYN.94 ',40('*'),/,
     1    51X,A8,2X,A8,/,1H1)
C
      LONCE=.FALSE.
  999 CALL INIT
      CALL VALUE
      CALL PLOUT(' ')
      CALL PLOUT( )
C
      WRITE(2,100)
  100 FORMAT(' ENTER 0 TO STOP, 1 TO REINITIALIZE, 2 '
     1    'FOR NO INIT PRINT')
      CALL BELL(1)
      READ(1,110) ISEC
  110 FORMAT(I1)
      IF(ISEC.NE.0) GO TO 999
      STOP
      END
```

```
      SUBROUTINE VALUE
      INTEGER LOW(2),HIGH(2)
      REAL MO, LF, LO, M1, LF1, L1, NORM1, LMAX, LMIN, MAXLF,
     1   NORMAX, LOSMAX, LF2, L2, NORM2, LAMDA1, LAMDA2,
     2   NORMG
      DIMENSION ACDM(100,2)
      LOGICAL PRTED, FLAG1, FLAG2
      INTEGER CUTIME
      COMMON /AMP/ TO,Q0S0,ALFA0,CL0,D0,L0,G,M0,C01,C02,C03,
     1  RHO,BETA,CLAF0,S0,D01,D02,M1,CLAF1,S1,D11,RLMAX,
     2  D12,THRESH,RK11,RK12,Q1S1,ALFA1,CL1,D1,L1,ALREAS
     3  ,P1,TAU,TAFTB,TFLAG,Q2S2,ALFA2,CL2,D2,L2
      COMMON /PARM1/ JJ,KSTEP
      COMMON /PARM2/ VTHO,VTH1,DTH,TSTEP,ICPUTM
      COMMON /PARM3/ NO,IO
      COMMON /PARM4/ STEP
      COMMON /PARM5/ MANUVR
      COMMON /PARM9/ ISEC,DMIS1,DMIS2,PRTED
      COMMON /ARRAY1/ XO(18),XOIN(18)
      COMMON /CONTR/ ACLF(100),ACBA(100),ACTI(100),TCHG,LF,U0,MAXLF
      COMMON /PLVR/ PL(100,55),PMAX,PMIN,LMAX,LMIN,SPMAX,
     1   SPMIN,ALFMAX,ALFMIN,NORMAX,ZMAX,ZMIN,VMAX,VMIN,
     2   DMAX,DMIN,XMAX,XMIN,YMAX,YMIN,ANMAX,LOSMAX
      COMMON /MSL/ LF1,U1,PSY1,THETA1,LF2,U2,PSY2,THETA2,
     1   SYDT1,THEDT1,SYDT2,THEDT2
      COMMON /SINCOS/ TABSIN(360),TABCOS(360)
C
      EQUIVALENCE (GAMA,XO(5)),(SIGMA,XO(6))
      EQUIVALENCE (VO,XO(4)),(V1,XO(10)),(V2,XO(16))
C
      DATA TESTN /1.0E-06/, IEXIT/'X'/, IQUIK/'Q'/
C
      COSGAM=COS(GAMA)
      SINGAM=SIN(GAMA)
      COSSIG=COS(SIGMA)
      SINSIG=SIN(SIGMA)
      COSSY1=COS(PSY1)
      COSSY2=COS(PSY2)
      COSTH1=COS(THETA1)
      COSTH2=COS(THETA2)
      SINSY1=SIN(PSY1)
      SINSY2=SIN(PSY2)
      SINTH1=SIN(THETA1)
      SINTH2=SIN(THETA2)
C
      TA=0.0
      JJ=0
      DUM1=1.0E10
      DUM2=1.0E10
      FLAG1=.FALSE.
      FLAG2=.FALSE.
      LAMDA1=0.5
C
C     FLAG(I) INDICATES THAT MISSILE I ALREADY GOT AWAY
C
  100 CONTINUE
      IF(FLAG1) LAMDA1=0.0
      IF(FLAG2) LAMDA1=1.0
      LAMDA2=1.0-LAMDA1
      IVAL=50
```

```
      IPRINT=TA
      VCX=V0*COSGAM*COSSIG
      VCY=V0*COSGAM*SINSIG
      VCZ=V0*SINGAM
      V1X=V1*COS(XU(11))*COS(XU(12))
      V1Y=V1*COS(XU(11))*SIN(XU(12))
      V1Z=V1*SIN(XU(11))
      V2X=V2*COS(XU(17))*COS(XU(18))
      V2Y=V2*COS(XU(17))*SIN(XU(18))
      V2Z=V2*SIN(XU(17))
      VRELX1=VCX-V1X
      VRELY1=VCY-V1Y
      VRELZ1=VCZ-V1Z
      VRELT1=SQRT(VRELX1**2+VRELY1**2+VRELZ1**2)
      VRELX2=VCX-V2X
      VRELY2=VCY-V2Y
      VRELZ2=VCZ-V2Z
      VRELT2=SQRT(VRELX2**2+VRELY2**2+VRELZ2**2)
      DELX1=XU(1)-XU(7)
      DELY1=XU(2)-XU(8)
      DELZ1=XU(3)-XU(9)
      DELX2=XU(1)-XU(13)
      DELY2=XU(2)-XU(14)
      DELZ2=XU(3)-XU(15)
C
      DUM1=AMIN1(DUM1,DMIS1)
      DUM2=AMIN1(DUM2,DMIS2)
      DMIS1=SQRT(DELX1**2+DELY1**2+DELZ1**2)
      DMIS2=SQRT(DELX2**2+DELY2**2+DELZ2**2)
      DSV1=0.5*DMIS1/VRELT1
      DSV2=0.5*DMIS2/VRELT2
      STEP=AMIN1(TSTEP,DSV1,DSV2)
      DMIS=AMIN1(DMIS1,DMIS2)
      IF(JJ .GT.0) MANUVR=1
      LF=1.0
      ULAST=U0
      U0=0.0
      IF(MANUVR.NE.1) GO TO 300
C
      U0=ULAST
      LF=MAXLF
      ISEC=ISEC+1
      GX1 = -(SYDT1*SINSY1*COSTH1 + THEDT1*COSSY1*SINTH1)
      GY1 = THEDT1*COSSY1*COSTH1 - SYDT1*SINSY1*SINTH1
      GZ1 = SYDT1*COSSY1
      NORM1 = SQRT(GX1*GX1 + GY1*GY1 + GZ1*GZ1)
      GX2 = -(SYDT2*SINSY2*COSTH2 + THEDT2*COSSY2*SINTH2)
      GY2 = THEDT2*COSSY2*COSTH2 - SYDT2*SINSY2*SINTH2
      GZ2 = SYDT2*COSSY2
      NORM2 = SQRT(GX2*GX2 + GY2*GY2 + GZ2*GZ2)
      IF(NORM1.GE.TESTN.AND.NORM2.GE.TESTN) GO TO 200
C
C     NORM TOO SMALL, NO GUIDANCE PLANE, DO NOTHING YET
C
      LF=ACLF(ISEC-1)
      GO TO 290
C
  200 CONTINUE
      GX1=GY1/NORM1
      GY1=GY1/NORM1
```

```
      GZ1=GZ1/NORM1
      GX2=GX2/NORM2
      GY2=GY2/NORM2
      GZ2=GZ2/NORM2
C
      COEFA1=GZ1*COSGAM-GY1*SINGAM*SINSIG-GX1*SINGAM*COSSIG
      COEFA1=COEFA1*(LO+TO*SIN(ALFAO))
      COEFB1=GY1*COSGAM-GX1*SINGAM
      COEFB1=COEFB1*(LO+TO*SIN(ALFAO))
      COEFC1=((TO*COS(ALFAO)-DO)*(GZ1*SINGAM+GY1*COSGAM*SINSIG
     1   +GX1*COSGAM*COSSIG)) - GZ1*MO*G
C
      COEFA2=GZ2*COSGAM-GY2*SINGAM*SINSIG-GX2*SINGAM*COSSIG
      COEFA2=COEFA2*(LO+TO*SIN(ALFAO))
      COEFB2=GY2*COSGAM-GX2*SINGAM
      COEFB2=COEFB2*(LO+TO*SIN(ALFAO))
      COEFC2=((TO*COS(ALFAO)-DO)*(GZ2*SINGAM+GY2*COSGAM*SINSIG
     2   +GX2*COSGAM*COSSIG)) - GZ2*MO*G
C
C     CHECK EVERY ANGLE (BY DEGREE) WITHIN +/- RLMAX
C
      NBLK=1
      PRFMAX=0.0
      IUO=(UO*180./PI)+180.
      IF(IUO.GT.360) IUO=IUO-360
      IF(IUO.LT.1) IUO=IUO+360
      IROLL=RLMAX*STEP
      LOW(1)=IUO-IROLL
      IF(LOW(1).GE.1) GO TO 210
      NBLK=2
      LOW(1)=1
      LOW(2)=360+IUO-IROLL
      HIGH(2)=360
  210 CONTINUE
      HIGH(1)=IUO+IROLL
      IF(HIGH(1).LE.360) GO TO 220
      NBLK=2
      HIGH(1)=IUO+IROLL-360
      HIGH(2)=360
      LOW(2)=LOW(1)
      LOW(1)=1
  220 CONTINUE
      DO 230 IJK=1,NBLK
      ISTRT=LOW(IJK)
      IEND=HIGH(IJK)
      DO 230 IUO=ISTRT,IEND
      PERF1=COEFA1*TABCOS(IUO)+COEFB1*TABSIN(IUO)+COEFC1
      PERF2=COEFA2*TABCOS(IUO)+COEFB2*TABSIN(IUO)+COEFC2
      ABS1=ABS(PERF1)
      ABS2=ABS(PERF2)
      IF(FLAG1) ABS1=ABS2*2.0
      IF(FLAG2) ABS2=ABS1*2.0
      PRFMIN=AMIN1(ABS1,ABS2)
      IF(PRFMIN.LE.PRFMAX) GO TO 230
      IFLAG=IVAL
      IMAX=IUO
      PRFMAX=PRFMIN
      IF(ABS1.LE.ABS2) GO TO 230
      IFLAG=-IVAL
  230 CONTINUE
```

```
      U0=FLOAT(IMAX-180)*PI/180.
C
C     NOTE --- IFLAG IS POSITIVE FOR LAMBDA = 1, AND NEGATIVE
C        FOR LAMBDA = 0
C
      PL(ISEC,2)=PERF1
      PL(ISEC,4)=PERF2
      PL(ISEC,26)=IFLAG
      PMAX=AMAX1(PMAX,PERF1,PERF2)
      PMIN=AMIN1(PMIN,PERF1,PERF2)
C
C     CALCULATE THE BEST ACCELERATION WITHOUT GUIDANCE PLANE
C
      ACCX=((T0*COS(ALFA0)-D0)*(COSGAM*COSSIG))-
     1   ((L0+T0*SIN(ALFA0))*(SIN(U0)*SINGAM+COS(U0)*
     2   SINGAM*COSSIG))
      ACCY=((T0*COS(ALFA0)-D0)*(COSGAM*SINSIG))+
     1   ((L0+T0*SIN(ALFA0))*(SIN(U0)*COSGAM-COS(U0)*
     2   SINGAM*SINSIG))
      ACCZ=((T0*COS(ALFA0)-D0)*SINGAM)-(M0*G)+
     1   ((L0+T0*SIN(ALFA0))*COSGAM*COS(U0))
      ANORM=SQRT(ACCX*ACCX+ACCY*ACCY+ACCZ*ACCZ)
C
C     STORE PLOTTED VARIABLES
C
      PL(ISEC,14)=NORM1
      PL(ISEC,19)=ANORM/M0
      PL(ISEC,30)=GX1
      PL(ISEC,31)=GY1
      PL(ISEC,32)=GZ1
      PL(ISEC,33)=NORM2
      PL(ISEC,34)=GX2
      PL(ISEC,35)=GY2
      PL(ISEC,36)=GZ2
      NORMAX=AMAX1(NORMAX,NORM1,NORM2)
      ANMAX=AMAX1(ANMAX,ANORM)
C
  290 CONTINUE
      IF(PRTED) GO TO 300
      PRTED=.TRUE.
      CCT=CCT+TAFTB*IFLAG
      ICHG=TPRINT
      WRITE(2,310) TPRINT
      WRITE(6,310) TPRINT
  310 FORMAT(2X,'START MANEUVER AT T = ',F6.2)
      CALL BELL(3)
C
      SPMAX=AMAX1(PL(1,9),PL(1,10),PL(1,39))
      SPMIN=AMIN1(PL(1,9),PL(1,10),PL(1,39))
      ALFMAX=ALFA0
      ALFMIN=ALFA0
      ZMAX=AMIN1(X0(3),X0(9),X0(15))
      ZMIN=AMAX1(X0(3),X0(9),X0(15))
      XMAX=AMAX1(X0(1),X0(7),X0(13))
      XMIN=AMIN1(X0(1),X0(7),X0(13))
      YMAX=AMAX1(X0(2),X0(8),X0(14))
      YMIN=AMIN1(X0(2),X0(8),X0(14))
      VMAX=V0
      VMIN=V0
      DMAX=D0
```

```
      DMIN=DU
      LOSMAX=0.0
C
  300 CONTINUE
      ACDA(ISEC)=DU*180./PI
      ACLF(ISEC)=LF
      ACTI(ISEC)=TPRINT
      ACDM(ISEC,1)=DMIS1
      ACDM(ISEC,2)=DMIS2
C
      IF (MOD(JJ,KSTEP) .NE. 0) GO TO 477
      WRITE (2,400) TPRINT, DMIS1, DMIS2
      WRITE (6,400) TPRINT, DMIS1, DMIS2
  400 FORMAT (5X,'TIME = ',F10.3,2X,'DSEP1 = ',G12.3,2X,
     1    'DSEP2 = ',G12.3)
C
  477 CONTINUE
      IF((V0.LT.VTH0).OR.(V1.LT.VTH1).OR.(V2.LT.VTH1)) GO TO 510
      IF(DMIS.LT.DTH) GO TO 520
      IF(TA.LT.3.0) GO TO 480
      FLAG1=FLAG1.OR.(DMIS1.GT.DUM1)
      FLAG2=FLAG2.OR.(DMIS2.GT.DUM2)
      IF(FLAG1.AND.FLAG2) GO TO 540
  480 CONTINUE
      CALL INTBOX
      TA=TA+STEP
      JJ=JJ+1
      IF(ISEC.GE.IO) GO TO 530
      GO TO 100
C
  510 CONTINUE
      WRITE (2,515) TPRINT
      WRITE (6,515) TPRINT
  515 FORMAT(1X,'**   A/C OR MISSILE VEL. IS TOO LOW ',
     1 'AT TIME: ',F10.3,'    **',/)
      GO TO 600
C
C     HIT OCCURRED, PRINT OUT
  520 CONTINUE
      WRITE(2,525) TPRINT,DUM1,DUM2
      WRITE(6,525) TPRINT,DUM1,DUM2
  525 FORMAT(1X,'*****  HIT  AT TIME = ',F10.3,
     1  '  *****',/,5X,'BEST DSEPS WERE 1:',
     2   G15.5,', & 2:',G15.6,/)
      GO TO 600
C
  530 CONTINUE
      WRITE(2,535) TPRINT
      WRITE(6,535) TPRINT
  535 FORMAT(5X,'TIME LIMIT AT T = ',F6.2)
      GO TO 600
C
C     CLOSURE RATE NEGATIVE SOLUTION -- PRINT IT
  540 CONTINUE
      WRITE(2,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
      WRITE(6,544) TPRINT,DUM1,DMIS1,DUM2,DMIS2
  544 FORMAT(1X,'**** CLOSURE RATE NEGATIVE AT TIME = ',F10.3,
     1   '****',/,5X,'TA1 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/,
     2    5X,'TA2 : BEST DSEP = ',G15.6,', NOW = ',G15.6,/)
      GO TO 600
```

```
C
  600 CONTINUE
      CALL BELL(1)
C
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      N10=6
      DO 620 J=1,N10
      JJJ=J+6
      KK=J+12
      TEMP=XO(J)
      TEMP1=XO(JJJ)
      TEMP2=XO(KK)
      IF(J.LT.5) GO TO 605
      TEMP=TEMP*180./PI
      TEMP1=TEMP1*180./PI
      TEMP2=TEMP2*180./PI
  605 CONTINUE
      WRITE(2,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
      WRITE(6,610) J,TEMP,JJJ,TEMP1,KK,TEMP2
  610 FORMAT(2X,'XO(',I1,'): ',G13.4,4X,
     1   'XO(',I2,'): ',G13.4,4X,
     2   'XO(',I2,'): ',G13.4)
  620 CONTINUE
C
      WRITE (2,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
      WRITE (6,625) DELX1,DELY1,DELZ1,DMIS1,DUM1
  625 FORMAT(/,10X,'DELX1: ',G12.3,2X,'DELY1: ',G12.3,/,
     1  10X,'DELZ1: ',G12.3,2X,'DMIS1: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      WRITE (2,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
      WRITE (6,630) DELX2,DELY2,DELZ2,DMIS2,DUM2
  630 FORMAT(/,10X,'DELX2: ',G12.3,2X,'DELY2: ',G12.3,/,
     1  10X,'DELZ2: ',G12.3,2X,'DMIS2: ',G12.3,/,
     2  10X,'BEST DMIS WAS ',G12.3)
      CALL BELL(1)
      READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
  635 FORMAT(A1)
C
      DO 650 I=1,ISEC
      IF(MOD(I,20).EQ.0) CALL BELL(1)
      IF(MOD(I,20).EQ.0) READ(1,635) LOGIC
      IF(LOGIC.EQ.IEXIT) RETURN
      IF(LOGIC.EQ.IQUIK) GO TO 660
      WRITE(2,636) ACTI(I),ACBA(I),PL(I,8),PL(I,38)
  636 FORMAT(1X,'TIME ',F5.2,3X,'ACBA ',F8.2,3X,'LF1C ',F8.1,
     1    3X,'LF2C ',F8.1)
  650 CONTINUE
  660 CONTINUE
      WRITE(6,641)
  641 FORMAT(1H1)
      DO 675 I=1,ISEC
      WRITE(6,637) ACTI(I),ACLF(I),ACBA(I),PL(I,8),PL(I,38),
     1    PL(I,17),PL(I,14),PL(I,11),PL(I,20),
     2    (ACOM(I,J),J=1,2),PL(I,47)
  637 FORMAT(1X,'TIME = ',0PF6.2,3X,'ACLF = ',F5.1,3X,
     1 'ACBA(DEG) = ',F8.2,3X,'LF1C = ',F8.1,3X,
     2    'LF2C = ',F8.2,3X,'NORM(A) = ',G11.2,/,20X,
```

```
     3   'NORM1(U) = ',G11.2,3X,'ALPHA = ',F5.2,3X,
     4   'DRAG = ',G11.2,/,20X,'DMIS1 = TA(1) = ',
     5   F9.2,5X,'DMIS2 = TA(2) = ',F9.2,5X,
     6   'LAMBDA1 = ',-2PF5.2,/)
  670 CONTINUE
C
      IDUM=CUTIME(J)-ICPUTM
      WRITE(2,639) IDUM
      WRITE(6,639) IDUM
  639 FORMAT(4X,'CPU TIME (SECONDS) = ',I5,0PF5.0)
C
      RETURN
      END
```